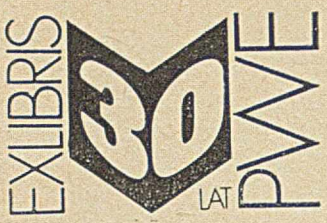
P. 1877/80

POLITECHNIKA ŚLĄSKA BIBLIOTEKA

2 1980

informatyka

WYDAWNICTWO NOT SIGMA



30 lat Państwowego Wydawnictwa Ekonomicznego

W marcu 1949 r. powołana została do życia jedna z największych obecnie oficyn wydawniczych — Państwowe Wydawnictwo Ekonomiczne, do 1961 r. funkcjonująca pod nazwą Polskiego Wydawnictwa Gospodarczego „Polgos". PWE wydało łącznie ok. 3500 książek w nakładzie ponad 23 miliony egzemplarzy oraz ok. 140 mln egz. czasopism. Wartość produkcji przekroczyła 1 mld zł.

W początkowym okresie działalności Wydawnictwo dostarczało przede wszystkim literatury typu poradnikowego i instruktażowego, niezbędnej dla ówczesnej kadry pracowników gospodarki oraz podręczniki dla szkół zawodowych. Już wtedy jednak pojawiły się ambitniejsze, istotne pozycje, takie jak np. prace K. Secomskiego, O. Langego, S. Szulca czy J. Pajestki.

W latach 1950—53 ukazała się nakładem PWE pierwsza w Polsce Ludowej wielotomowa publikacja typu encyklopedycznego — „Słownik towaroznawstwa", który w ośmiu tomach objął ponad 12 tys. haseł. Z późniejszych publikacji tego typu na czoło wysuwa się „Encyklopedia ekonomiczna". Obok niej zaś warto wymienić — encyklopedie: statystyki i rachunkowości, słowniki: matematyki i cybernetyki, handlowo-ekonomiczny angielsko-polski i polsko-angielski, towaroznawczy.

Od 1956 r. do dzisiaj włącznie wydawane są takie pozycje, jak Rocznik Polityczny i Gospodarczy czy Mały Słownik Ekonomiczny. W tym czasie rozpoczęto też kilka znaczących serii wydawniczych. Obok największych (Biblioteka Gospodarki Magazynowej, Wszystko o gospodarce, Polska — RWPG — Świat) pojawiła się także seria „informatyczna" pn. Informatyka w Praktyce, realizowana od 1975 r. W jej ramach wydano do tej pory 16 pozycji.

Seria ta poświęcona jest popularyzowaniu zastosowań informatyki, a zwłaszcza problemom budowania i funkcjonowania systemów informatycznych w jednostkach gospodarczych oraz doskonalenia sposobów wykorzystywania metod informatycznych w zarządzaniu.

W latach 1968—77 Wydawnictwo opublikowało 36 pozycji z zakresu informatyki, co stanowiło 19% wszystkich 231 publikacji polskich na ten temat. W ciągu ostatnich dwóch lat pojawiło się niespełna 20 następnych pozycji. Charakterystyczna jest analiza prac dotyczących informatyki, wydanych we wspomnianej dekadzie. Z 36 pozycji aż 21 poświęconych było zastosowaniom, 8 — oprogramowaniu, 6 — zagadnieniom ogólnym i tylko jedna problemom sprzętu. 31 książek podpisali autorzy polscy, pozostałą zaś część stanowiły przekłady.

PWE jest poza tym głównym edytorem podręczników akademickich dla wyższych szkół ekonomicznych; między innymi z zakresu informatyki (np. „Wstęp do informatyki" prof. T. Pechego).

Nie sposób tu — oczywiście — wymienić wszystkich publikacji PWE, które na wyróżnienie zasługują. Szeroki zestaw prac polskich autorów, które odbiły się echem w literaturze światowej, uzupełniony został niemałym wyborem dzieł zagranicznych, zarówno z krajów naszego obozu, jak też i z innych krajów świata. Celem Wydawnictwa jest — jak samo twierdzi — podążanie za postępem i rozwojem poszczególnych dyscyplin. Jednym z nowych obszarów zainteresowań, zgodnie z tendencjami światowymi, jest w PWE — informatyka ekonomiczna.

W sferze popularyzacji osiągnięć nauki i praktyki, w procesie doskonalenia zawodowego i wymiany doświadczeń, należy też uwzględnić niebagatelną rolę czasopism, wydawanych przez PWE. Są one środkiem szybkiego przekazu myśli, dzięki czemu dają szansę pobudzenia czytelników do wnikliwej dyskusji i tym samym — do uczenia się ekonomii w niekonwencjonalny sposób. Stosunkowo wysokie nakłady tych czasopism wskazują na szeroki zakres ich oddziaływania.

Książki i czasopisma PWE, cieszące się niewątpliwie uznaniem licznej rzeszy czytelników, służą ekonomicznej edukacji społeczeństwa. Potwierdzeniem wartości merytorycznej publikacji Państwowego Wydawnictwa Ekonomicznego są dziesiątki nagród i wyróżnień, w tym — Polskiej Akademii Nauk.

WYDAWNICTWO
NACZELNA ORGANIZACJA TECHNICZNA
SIGMA
CZASOPISM I KSIĄŻEK TECHNICZNYCH
ul. Świętokrzyska 14a
00-950 Warszawa
skrytka pocztowa 1004

KOLEGIUM REDAKCYJNE

Redaktor naczelny: prof. dr hab. Leon ŁUKASZEWICZ

dr Krystyn BERNATOWICZ, prof. dr hab. inż. Konrad FIAŁKOWSKI (zastępca redaktora naczelnego), dr Janusz GWIAZDA, dr inż. Marek HOŁYŃSKI, mgr inż. Stanisław JASKÓLSKI, Władysław KLEPACZ (zastępca redaktora naczelnego), mgr inż. Wincenty ŁADA, mgr Stanisław MROZIK, dr inż. Tomasz PAWLAK

Sekretarz redakcji: Anna GLUTH-NOWOWIEJSKA Red. techn.: Ewa KAMIŃSKA

RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BAŃKOWSKI (sekretarz), mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI, mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon ŁUKASZEWICZ, gen. dr inż. Marian PASTERNAK, mgr inż. Bronisław PIWOWAR, mgr Zbigniew SUBSTYK, mgr Jerzy TRYBULSKI, doc. dr hab. Tadeusz WALCZAK, dr inż. Jan ŻYDOWO

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 381, tel. 27-71-40 lub centrala 26-82-61 w. 285, dyżury redakcji 10.00—13.00

Zakł. Graf. „Tamka". Zam. 582. Papier druk. sat. V kl. 70 g. A1. Obj. 5 ark. druk. Nakład 7400 egz. O-55.

Cena egzemplarza zł 30.— INDEKS 36124 Prenumerata roczna zł 360.—

Informatyka

zastosowania w gospodarce, technice i nauce

Nr 2
MIESIĘCZNIK
1980
ROK XV
luty

ORGAN KOMITETU INFORMATYKI, MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO I TECHNIKI ORAZ KOMITETU NAUKOWO-TECHNICZNEGO NOT DS. INFORMATYKI

P. 1877/80

W NUMERZE:

Strona

Z myślą o informatyce lat osiemdziesiątych
Jan Kramarczuk 2

Informatyka a społeczeństwo. Część 2
Władysław M. Turski 4

Współczesne metody projektowania systemów informatycznych
Marek Greniewski 9

Praktyczne aspekty stosowania metod strukturalnych
Elżbieta Niemyjska 11

CAMAC — blokowy system elektroniczny do automatyzacji pomiarów i sterowania
Stanisław Koślacz, Krzysztof Rzymkowski 14

Doświadczenia z programowania w języku CSL ODRA 1300
Antoni Machowski, Wiesław Starowicz 17

Metody heurystyczne w przetwarzaniu danych
Bogdan Stefanowicz 20

CURRICULUM'78 — program kształcenia informatyków w szkołach wyższych
oprac. Jan Bobrowski 22

ZE ZJEDNOCZENIA INFORMATYKI

XV lat ZETO Wrocław *(W.K.)* 25
Uroczystość jubileuszowa *(K.B.)* 26

USPRAWNIENIA, NOWE KONCEPCJE, POMYSŁY

System CAMAC w sterowaniu robotem przemysłowym
Andrzej Gogolewski, Adam Żukowski 27

CENTRUM ETOB

Odrowski remanent
Wincenty Łada 29

PORTRETY ZAWODOWE
Gdzie diabeł nie może... *(wład)* 30

TRYBUNA CZYTELNIKA

(Hiper)wtajemniczenie teoriografowe
Maciej M. Sysło 31

Z KRAJU

Komputeryzacja wielkich miast — WARSZAWA *oprac. Z. G.* 32
„Matematyka bliżej przemysłu, przemysł bliżej matematyki" *(ABE)* 33

NASZE RECENZJE

W poszukiwaniu rozwiązań optymalnych
Mieczysław Muraszkiewicz 34

ZE ŚWIATA

Międzynarodowe konferencje 35

PROBLEMATYKA BAZY DANYCH

Problemy budowy wspólnych baz danych
Grzegorz Gruchman 36

Regeneracja baz danych
Leszek A. Maciaszek 38

Cz.3573

W niniejszym numerze zamieszczamy kolejne referaty (lub ich obszerne fragmenty) wygłoszone na IV Krajowej Konferencji Informatyków, która odbyła się w dniach 22—24 października 1979 r. we Wrocławiu.

JAN KRAMARCZUK
Rządowe Centrum Informatycznego
Systemu Planowania Centralnego
Warszawa

Z myślą o informatyce lat osiemdziesiątych

Efektywne wykorzystanie coraz większej ilości informacji we wszystkich obszarach ludzkiej działalności staje się podstawowym problemem końca XX wieku. Problem ten może rozwiązać tylko informatyka.

Obecnie zainstalowanych jest na świecie ok. 640 tysięcy komputerów oraz ok. 2 milionów terminali o łącznej wartości ponad 100 miliardów dolarów. Przemysł informatyczny stwarza coraz większe możliwości zaspokojenia potrzeb użytkowników. Zwiększa się niezawodność sprzętu, maleją koszty obliczeń. Ulepszony sprzęt telekomunikacyjny oraz odpowiednie oprogramowanie umożliwiają budowę sieci komputerowych, stosowanie rozproszonego przetwarzania danych. Kształtuje się nowa infrastruktura informacji — informację statyczną zastępuje informacja dynamiczna, wspomagana przez modele ekonomiczno-matematyczne.

W zachodniej Europie, Stanach Zjednoczonych i Japonii powstały już i w dalszym ciągu powstają — liczne sieci transmisji danych. W krajach RWPG także poświęca się dużo uwagi rozwojowi systemów sieciowych, szczególnie w Związku Radzieckim — z uwagi na olbrzymie potrzeby i możliwości.

Jednym z warunków zwiększenia efektywności zarządzania w naszym kraju jest **pogłębienie współpracy z krajami RWPG.** Realizowane są już wieloletnie plany współpracy i wymiany doświadczeń. Wymiana doświadczeń z Głównym Centrum Obliczeniowym GOSPLANU w ZSRR obejmuje m.in. zastosowanie komputerów do planowania zasobów, do oceny zapotrzebowania gospodarki narodowej na surowce energetyczne i metale. Współpraca z centrum informatyki Komisji Planowania CSRS obejmuje zagadnienia dotyczące systemu informacji kierownictwa SIK, handlu zagranicznego oraz wymianę doświadczeń w zakresie eksploatowanej bazy technicznej. Przedmiotem współpracy z WRL są dynamiczne modele gospodarki narodowej na potrzeby planowania perspektywicznego, natomiast we współpracy z ośrodkiem informatycznym Komisji Planowania Rumunii rozwiązywane są zagadnienia dotyczące rozwoju banków danych.

Możliwości wynikające ze współpracy krajów RWPG wymagają, aby poświęcono jej więcej uwagi — zarówno w zakresie wymiany doświadczeń, jak i aktywnego uczestnictwa w pracach Komisji Międzyrządowej ds. ETO oraz we wspólnych działaniach dotyczących rozwiązywania technicznych problemów bazy normatywnej. Bardzo istotnym czynnikiem dla rozwoju informatyki jest też kierunek współpracy przyjęty przez RWPG — specjalizacja, obejmująca produkcję sprzętu i oprogramowania, metodykę zastosowań w poszczególnych obszarach zarządzania, nowoczesne techniki projektowania i programowania.

Chciałbym podkreślić, że podejmując problematykę informatyczną lat osiemdziesiątych należy zdać sobie sprawę ze sztuczności podziałów czasowych, ponieważ to, co się będzie działo w nadchodzących latach, jest przecież zawarte w zagadnieniach, którymi zajmujemy się obecnie — zmiany tradycyjnych struktur organizacyjnych, jakość sprzętu komputerowego i możliwości jego szerszego zastosowania, wdrażanie nowoczesnych, strukturalnych technik projektowania i programowania, rozwój kadry informatycznej, szkolenie użytkowników.

Nowe generacje układów elektronicznych pojawiają się co 6—7 lat, przynosząc wielokrotny wzrost prędkości liczenia, pojemności pamięci, a przede wszystkim niezawodności sprzętu. Coraz częściej pojawiają się minikomputery. Istotną rolę w latach osiemdziesiątych odegrają z pewnością **systemy sieciowe,** co stworzy szczególnie dogodne warunki dla rozwoju systemów informacji kierownictwa. Zmiany zachodzące w oprogramowywaniu aplikacyjnym będą miały duży wpływ na rozwój sprzętu. Pisanie szczegółowych procedur określających kolejne etapy pracy komputera zastąpią języki proceduralne.

Odmienny rodzaj oprogramowania zwanego nieproceduralnym reprezentują **wyspecjalizowane pakiety aplikacyjne.** W krajach zachodnich pojawia się ich coraz więcej; są bardzo zróżnicowane — co umożliwi w przyszłości szeroki zakres zastosowań. Istotą ich jest możliwość używania języka potocznego bez znajomości programowania komputera. Jest to olbrzymi skok w technice programowania, umożliwiający nie tylko masowe zastosowania, ale również radykalne zmniejszenie kosztów przetwarzania. Prognozy przewidują, że w połowie lat osiemdziesiątych kraje zachodnie ponad połowę czasu komputera będą przeznaczać na programy aplikacyjne zbudowane w językach nieproceduralnych. Ten kierunek rozwoju musimy uwzględnić i w naszych planach.

Potrzeby użytkowników podlegają oddziaływaniu dwóch, częściowo przeciwnych sobie czynników: dążeniu do ekonomiczności i dążeniu do szerokiego zastosowania informatyki. Nie zawsze umiemy powiedzieć, co jest ekonomiczne oraz określić w postaci jakościowej korzyści wynikające z zastosowań komputerów. Ocena korzyści sprowadza się najczęściej do oszacowania efektu wyeliminowania czynników hamujących rozwój gospodarki. Ocena taka może być przeprowadzona tylko subiektywnie. Efekty ekonomiczne rozwoju informatyki daje się także przedstawić w ujęciu ilościowym, np. lepsze wykorzystanie kadr w zakładzie pracy, racjonalne kierowanie potencjałem przedsiębiorstwa transportowego, wybór optymalnego rozwiązania wielu zadań w skali całej gospodarki narodowej. Z punktu widzenia racjonalnego gospodarowania potencjałem sprzętowym informatyki często rozważany jest związek między wprowadzaniem techniki obliczeniowej a zmniejszeniem nakładu siły roboczej. Obecnie nie jest to jeszcze najbardziej charakterystyczny wskaźnik korzyści ekonomicznych informatyzacji. Natomiast w następnych latach ocenę taką będą stanowiły efekty ekonomiczne, uzyskane dzięki informacjom ułatwiającym podejmowanie decyzji.

Chciałbym teraz omówić **główne kierunki rozwoju zastosowań** informatyki w nadchodzącym dziesięcioleciu. Przede wszystkim będą to w dalszym ciągu zastosowania w sterowaniu procesami technologicznymi, zarządzaniu, pracach inżynierskich. Nie należy jednak zapominać, że niezwykle szybko wzrasta zapotrzebowanie na **usługi.** Punkt ciężkości, skoncentrowany obecnie na produkcji dóbr materialnych, powinien więc przesunąć się w kierunku usług. Czy uda się nam osiągnąć w dziedzinie usług taką samą wydajność, jakiej dopracowaliśmy się w przedsiębiorstwach przemysłowych? Nie chciałbym być nadmiernym optymistą, ale uważam, że możliwości zastosowania informatyki w usługach są olbrzymie i — realne. Ludzie związani z usługami zdają sobie sprawę z konieczności zwiększenia wydajności pracy, zaczynamy też poznawać mechanizm, za pomocą którego da się ten cel zrealizować. Wierzę, że zastosowanie informatyki w usługach, zwłaszcza w łączności, transporcie, szkolnictwie, handlu i służbie zdrowia, nadal będą miały priorytet.

Drugi nurt prac, których realizację umożliwi rozwój systemów sieciowych, **to zastosowania informatyki na potrzeby władz terenowych.** Ich celem jest usprawnienie zarządzania i systemu informacji dla szczebla centralnego, pochodzących zarówno z terenowych organów administracji państwowej, jak i jednostek gospodarczych zarządzanych centralnie. Potrzeby informacyjne terenowych organów administracji państwowej są w pewnym zakresie zaspokajane przez systemy rządowe, niektóre resortowe oraz wycinkowe własne systemy ewidencyjne. Zbiory informacji źródłowych, jakimi dysponują urzędy wojewódzkie, są w zasadzie wystarczające, natomiast operowanie nimi, selekcja i udostępnianie do celów zarządzania wymagają wiele pracy, zarówno organizacyjnej, jak i projektowo-wdrożeniowej, ale przede wszystkim wymagają odpowiednich nakładów.

Doceniając wagę zagadnienia decyzją Prezesa Rady Ministrów z kwietnia 1978 r. powołano zespół, który miał dokonać analizy stanu zastosowania informatyki w terenowych organach administracji państwowej i przedstawić propozycje dalszego postępowania w tym zakresie. W opracowanym raporcie w pierwszej kolejności zaproponowano obszary zastosowań, obejmujące wszystkie województwa:
— ewidencja gruntów
— ewidencja jednostek gospodarczych
— ewidencja indywidualnych i zespołowych gospodarstw specjalistycznych
— gospodarka środkami trwałymi
— ewidencja budynków i lokali mieszkalnych
— rozliczanie czynszów
— ewidencja placówek służby zdrowia
— ewidencja pojazdów samochodowych
— planowanie i nadzór nad funkcjonowaniem komunikacji w miastach
— sprawozdawczość budowlana.

Ze względu na duży stopień złożoności systemów ewidencyjnych i koszty związanę z ich realizacją decyzje o wdrażaniu tych systemów należy uzależnić od możliwości kadrowych, finansowych i technicznych w danym województwie. Proponuje się, aby wojewódzkie systemy ewidencyjne były wdrażane we wszystkich województwach według zasad ustalonych przez Ministerstwo Administracji, Gospodarki Terenowej i Ochrony Środowiska, w oparciu o jednolity system identyfikacji terenowej. Jest to przedsięwzięcie wymagające sprawnej organizacji oraz odpowiednich środków — zarówno kadrowych, jak i technicznych.

Ewidencjonowanie stanu infrastruktury społeczno-technicznej i kontroli realizacji w sferze działalności społeczno-gospodarczej (rejestrowanie, sprawdzanie, agregowanie, sporządzanie sprawozdań) jest sprawą istotną, ale stanowiącą jedynie punkt wyjścia dla systemów planistycznych, prognostycznych oraz analiz zjawisk przyczynowo-skutkowych.

Przewidywany efekt zastosowań to przede wszystkim:
— zapewnienie pełnego zestawu informacji na potrzeby zarządzania w terenowych organach administracji państwowej
— uporządkowanie ewidencji w podstawowych dziedzinach gospodarki terenowej
— przyspieszenie wymiany informacji pomiędzy gminą a województwem
— zwiększenie możliwości kontroli zadań społeczno-gospodarczych
— stworzenie korzystniejszych warunków dla usprawnienia prac planistycznych
— zwiększenie kontroli realizacji planów społeczno-gospodarczych
— przygotowanie kadry i organizacji urzędów wojewódzkich do wykorzystania nowoczesnych technik — informatyki.

Coraz bardziej złożone stają się procesy zarządzania gospodarką, szybkie zmiany zachodzą w konfiguracji sprzętu i architekturze, toteż konieczne stają się systematyczne **prace naukowo-badawcze** oraz nieustanna konfrontacja bieżących zamierzeń i osiąganych wyników z możliwościami, które daje aktualny poziom wiedzy oraz zdobyte doświadczenie.

Wiele placówek naukowo-badawczych i wyższych uczelni podjęło już odpowiednie prace, zastosowania jednak w dalszym ciągu wyprzedzają poważnie opracowania teoretyczne. Wpływa to ujemnie na jakość rozwiązań praktycznych. NPSG przewiduje wysokie nakłady na rozwój nauki i techniki, a przede wszystkim na rozwój badań umożliwiających wdrożenie nowych opracowań i technologii zrealizowanych w krajowych placówkach naukowo-badawczych. Większość priorytetowych problemów w obszarze informatyki rozwiązywanych jest w ramach funkcjonującego od 10 lat systemu problemów węzłowych, umożliwiającego koncentrację wysiłku kadry naukowo-badawczej w rozwiązywaniu zagadnień począwszy od prac podstawowych do wdrożenia.

Prace badawcze i rozwojowe są kosztowne i ryzykowne. Istotnym elementem jest więc stworzenie klimatu, który sprzyjałby procesom innowacyjnym i nagradzał badania oraz innowacje.

Aby usprawnić badania nad zastosowaniem informatyki, a przede wszystkim wykluczyć podejmowanie tematów, które już zostały częściowo bądź całkowicie rozwiązane, należy przede wszystkim doprowadzić do opracowania i ciągłego aktualizowania rejestru oprogramowania użytkowego — wdrożonego lub znajdującego się w fazie prac projektowych. Poza tym konieczne są równolegle prowadzone badania dotyczące przyszłych potrzeb użytkowników. Prace takie przyspieszyłyby realizację zamierzonego celu, a przede wszystkim ograniczyły nakłady na prace badawcze, miałyby też duży wpływ na centralizację planowania rozwoju oprogramowania.

Zagadnienie, które powszechnie jest znane, a które jak dotychczas zbyt powierzchownie traktowano przede wszystkim w fazie realizacji, to wspólne prace naukowo-badawcze nad oprogramowaniem aplikacyjnym w ramach obozu socjalistycznego. Obecnie współpraca ogranicza się jedynie do konferencji i spotkań w ramach równoległych instytucji. Tymczasem obóz socjalistyczny skupia największy potencjał matematyczny na świecie, ma też bogate doświadczenia. Wspólny wysiłek przyniósłby poważne efekty.

* * *

Wszyscy wiemy, jak istotny jest wpływ masowego zastosowania nowych środków technicznych na poglądy ludzi, jakość życia i rozwój ogólnospołeczny. Wystarczy przypomnieć historię maszyny parowej, elektrycznych silników i generatorów lub — w nieodległej przeszłości — rozpowszechnianie silników spalinowych. We współczesnym świecie coraz silniej zarysowuje się wpływ informatyki.

Wprowadzenie informatyki do programu szkół średnich, pomaturalnych i wyższych umożliwiło stworzenie nowej kadry specjalistów — i to nie tylko informatyków, ale i ludzi, których działalność zawodowa wiąże się z innymi obszarami działalności gospodarczej. Nowy pogląd na technikę informatyczną kształtuje więc nowe pokolenie.

Zmiany w poglądach w ciągu najbliższych 10—15 lat oznaczają szerokie zastosowanie informatyki w krajach przede wszystkim o rozwiniętym przemyśle. Jestem przekonany, że zastosowanie prostej algorytmizacji czy komunikacji z ośrodkami obliczeniowymi w celu otrzymania potrzebnej informacji stanie się w bliskiej przyszłości sprawą zwykłą i powszechną.

WŁADYSŁAW M. TURSKI
Instytut Informatyki UW
Warszawa

Informatyka a społeczeństwo. Część 2

W zastosowaniach informatyki w zarządzaniu można wyróżnić trzy nurty: sprawozdawczość, automatyzację operacji finansowo-księgowych i przetwarzanie informacji do celów operacyjno-decyzyjnych. Podział ten nie jest ściśle rozłączny, ale pomoże nam w skupieniu uwagi na głównych problemach rozważanej dziedziny zastosowań.

Działalność sprawozdawcza, towarzysząca rozwojowi człowieka od powstania jakiejkolwiek cywilizacji, zawsze była procesem informacjochłonnym. Głównym jej celem jest gromadzenie wiadomości używanych przy formułowaniu celów strategicznych świadomej działalności politycznej i organizacyjnej społeczeństw oraz ich autonomicznych składowych, a także — aczkolwiek następuje to dopiero w rozwiniętych stadiach cywilizacji przemysłowej — wiadomości używanych przy wytyczaniu celów gospodarczych. Relatywnie późniejsze występowanie tej funkcji sprawozdawczości wynika z tego, że struktury organizacyjne społeczeństwa zdolne do świadomego formułowania polityki gospodarczej wykształcają się dosyć późno w historycznym rozwoju cywilizacji przemysłowej. Zauważmy też, że funkcjonowanie struktur formułujących zasady polityki gospodarczej pozostaje w sprzeczności z podstawowym kodeksem praw osobistych w rozumieniu burżuazyjnym i do dzisiaj znajduje się w ogniu nieustannej krytyki w wysoko rozwiniętych krajach kapitalistycznych, spotykając się nawet z czynnym oporem nie tylko wielkich przedsiębiorstw, lecz także odpowiednio pobudzanych ruchów społecznych [15].

Jeśli w uproszczeniu przyjąć, że organizatorem sprawozdawczości jest państwo, to można zauważyć ścisłe korelacje zakresu sprawozdawczości użytecznej (tj. takiej, której wyniki są społecznie użyteczne) ze społecznie akceptowanymi funkcjami, które państwo pełni. Państwa cywilizacji rolniczej mogą przeto wprowadzić sprawną i powszechną sprawozdawczość wykorzystywaną w celach obrony narodowej (prowadzenia wojen) [16], ale napotykają już pewne opory przy wprowadzeniu sprawozdawczości gospodarczej, mogącej służyć za podstawę polityki podatkowej — jedynej formy ingerencji tych państw w sferę ekonomiczną. Zresztą, im mniej absolutystyczne są zasady sprawowania władzy w takich społeczeństwach, tym trudniej prowadzić działalność sprawozdawczą, a tym bardziej — wykorzystywać ją w celach gospodarczych (exemplum: paraliżująca działalność sejmów Rzeczypospolitej szlacheckiej).

W krajach socjalistycznych, gdzie polityczne i gospodarcze role organizacji państwowej ulegają w dużym stopniu połączeniu, łatwość egzekwowania postanowień wprowadzających czynności sprawozdawcze stwarza pozory możliwości objęcia sprawozdawczością wszystkich aspektów życia społecznego i iluzję, że obfitość tą drogą uzyskanych wiadomości sama z siebie wpłynie na skuteczność strategicznych decyzji gospodarczo-społecznych. Pojawiają się przy tym dwie tendencje, wynikające bardziej z psychologicznych własności człowieka niż z przesłanek obiektywnych. Chodzi o dążność do zaspokajania ciekawości i ufność w to, że „wiedzieć znaczy móc działać".

Komputeryzacja procesów sprawozdawczych stwarza istotnie niebywałe możliwości zaspokajania ciekawości, które nie temperowane świadomością tego, co obiektywnie wiedzieć potrzeba, prowadzą do rakowatego rozrostu sprawozdawczości [17]. Rozbudzonej i łatwo zaspokajanej ciekawości towarzyszy zazwyczaj nieufność względem danych zagregowanych, syntetycznych, podejrzewanych o błędy wynikające z niepełnej analizy i syntezy. Nie dostrzega się w praktyce nieuchronnych błędów danych jednostkowych, które dopiero przy agregowaniu (jeśli nie są to błędy systematyczne) pozwalają uzyskać obraz prawidłowy, aczkolwiek o mniejszej zdolności rozdzielczej. Zjawiska te, niestety trudne do wykorzenienia, nie są skutkiem komputeryzacji, ale przy komputeryzacji występują ze wzmożoną siłą. Stronę psychologiczną sprawy dość łatwo można pojąć; jest to kompleks małego kaprala, który potrafi zwrócić się do rannego po imieniu i nazwisku, wspominając przy tym wspólnie toczone bitwy. Wielu działaczy uważa, że znajomość szczegółów świadczy o dobrym opanowaniu intelektualnym przedmiotu. Odnoszę wrażenie, że jednym z celów rozbudowy sprawozdawczości jest chorobliwa chęć „wiedzenia" wszystkiego w zaskakujących szczegółach.

Odczucia społeczne wobec galopującej sprawozdawczości są na ogół negatywne, niechęć potęguje się jeszcze przy zbyt oczywistej komputeryzacji tej działalności, co może prowadzić nieraz do postaw wręcz wrogich. W różnych krajach obserwuje się reakcje przekory wobec skomputeryzowanej sprawozdawczości: próby „oszukania" komputera, złośliwe uszkodzenia kart informacyjnych przeznaczonych do automatycznego wczytywania, pełne Schadenfreude enuncjacje prasowe o pomyłkach komputera w tej właśnie dzie-

[15] Wyraźne tego przejawy można zaobserwować w ostatnich latach w USA, gdzie populistyczne hasła ograniczenia gospodarczej roli rządu federalnego nie tylko wprowadziły do Białego Domu reprezentanta tej ideologii (ze znanymi reperkusjami nawet na skalę międzynarodową), lecz także spowodowały takie ewenementy społeczno-ekonomiczne, jak uchwalenie słynnej „Propozycji 13" w referendum ludności stanu Kalifornia (ograniczenie bazy materialnej działalności rządu stanowego) czy ogromny wzrost efektywności działań przeróżnych ugrupowań „ochrony środowiska" (np. wstrzymanie konstrukcji kompleksu hydroenergetycznego, po zainwestowaniu weń kilkuset milionów dolarów, tylko dlatego, że ukończenie kompleksu spowodowałoby zagładę pewnego gatunku ryb, a raczej rybek, poza ichtiologami nikomu przed tą sprawą nie znanych). Pouczające jest w tej mierze referendum z listopada 1978 r., w którym ludność Austrii zabroniła uruchomienia elektrowni jądrowej w Zwentendorf, zbudowanej kosztem 500 mln dol., przy czym wielu obserwatorów twierdzi, że decydujący wpływ na przebieg referendum wywarło silne powiązanie przez kanclerza Kreisky'ego przedmiotu referendum z pryncypiami rządowej polityki ingerowania w sferę ekonomiczną

[16] Zasady ewidencji ludności wprowadzone przez królów szwedzkich w epoce wojny trzydziestoletniej przetrwały do dzisiaj w nie zmienionych strukturach organizacyjnych, mimo kilkakrotnej zmiany środków technicznych używanych do jej prowadzenia

[17] Pani komandor Grace Hopper, główna projektantka ogromnego systemu gromadzenia i przetwarzania informacji dla kwatermistrzostwa Marynarki Wojennej USA, skarżyła mi się kiedyś, że jedną z największych trudności przy realizacji tego projektu była walka z admirałami, dowódcami poszczególnych flot i komendantami teatrów działań, którzy koniecznie chcieli mieć możliwość uzyskiwania szczegółowych i detalicznych wiadomości o stanie magazynów, mimo że nie ma żadnego racjonalnego powodu, dla którego np. dowódca VI Floty, operującej na Morzu Śródziemnym, potrzebowałby wiedzieć, ile puszek soku pomarańczowego znajduje się na 5 półce 4 przegrody w 7 magazynie w Neapolu, nie mówiąc już o zapasach mydła w kantynie oficerskiej bazy na Okinawie. Ponieważ jednak system zawierał takie wiadomości, admirałowie koniecznie chcieli mieć możliwość łatwego i bezpośredniego ich uzyskania

dzinie. Jednocześnie pojawia się dążność do nałożenia na komputerową sprawozdawczość i działalność informacyjną ograniczeń nie stosowanych w przedkomputerowych formach tych czynności [18]).

Użyteczność skwapliwie gromadzonych informacji sprawozdawczych zależy od dwu czynników: wiarygodności danych źródłowych (a przynajmniej braku błędów systematycznych) i możliwości sprawnego korelowania zgromadzonych wiadomości. Warto przy tym pamiętać, że czynność korelowania zgromadzonej informacji wymaga znacznie większej mocy obliczeniowej niż jej gromadzenie. Wynika to stąd, że gromadzenie i przechowywanie informacji wymagają użycia mocy obliczeniowej proporcjonalnej do objętości informacji, podczas gdy wszechstronna korelacja (wszystkich wiadomości ze wszystkimi) jest procesem wymagającym użycia wykładniczej ilości mocy obliczeniowej. Jeśli nawet uwzględnić fakt, że moc obliczeniowa wzrasta z kwadratem nakładów na jej wytworzenie (prawo Groscha), dochodzimy do wniosku, o ile wiem nie uwzględnianego przy planowaniu przedsięwzięć w tej dziedzinie, że zamierzając n-krotne zwiększenie ilości informacji objętej sprawozdawczością, wystarczy zaplanować $\sqrt{n}$-krotne zwiększenie nakładów na sprzęt informatyczny używany do gromadzenia tej informacji, lecz dla wszechstronnego skorelowania zebranych wiadomości będą potrzebne nakłady podniesione do potęgi $n/2$. Powinno to być przestrogą przed zwiększeniem objętości sprawozdawczości bez jednoczesnego, z góry przewidywanego, znacznego agregowania uzyskiwanych informacji.

Tendencje do wzrostu sprawozdawczości, cechujące wszystkie społeczeństwa o silnej i scentralizowanej władzy państwowej, mające istotne prerogatywy gospodarcze (a nie są to tylko społeczeństwa socjalistyczne!), napotykają — oprócz wymienionych już przeszkód natury technicznej i psychologicznej — również przeszkodę ekonomiczną. Chodzi mianowicie o to, że ze względu na konieczność ręcznego przeniesienia wielkiej ilości danych na maszynowo-czytelne nośniki, wzrost sprawozdawczości wymaga odpowiedniego zwiększenia zatrudnienia, przy czym znane dotychczas urządzenia w niewielkiej tylko mierze zwiększają wydajność pracy; zauważmy: pracy wykonywanej podwójnie, bo i przy wypełnianiu kwestionariusza (formularza), i przy jego kodowaniu.

Im droższa jest praca ludzka, tym kosztowniejsze staje się przygotowanie danych dla nieokiełznanej sprawozdawczości. Im efektywniej jest wykorzystywana godzina pracy człowieka w przemyśle (skąd wypierają ją znacznie tańsze urządzenia informatyczne, wobec czego znajduje zastosowanie tylko tam, gdzie aktualnie nie może być zastąpiona programowanym automatem), tym wyraźniej widzimy tutaj dylemat: albo zatrudnieni przy wprowadzaniu danych pracownicy będą istotnie gorzej opłacani za jednostkę pracy niż pracownicy, którzy jeszcze pracują w przemyśle, albo sprawozdawczość musi przynieść tak znaczne korzyści ekonomiczne, aby „zarobiła" na wyższe płace rzesz zatrudnionych przy wprowadzaniu danych.

Pozostawiając na potem rozważania o tym, czy i w jakich warunkach sprawozdawczość może przynieść istotne korzyści ekonomiczne, zauważmy, że pierwsze rozwiązanie — a więc znaczne i powiększające się nierównomierności płacy godzinowej robotnika zatrudnionego przy przygotowywaniu danych i robotnika zatrudnionego przy obsłudze skomputeryzowanych urządzeń wytwórczych — jest na dłuższą metę nie do utrzymania, nawet jeśliby zatrudnienie przy przygotowywaniu danych miało być traktowane jako element redystrybucji rezerw siły roboczej. Należy bowiem pamiętać, że dyslokacji podlegają robotnicy przemysłowi, którzy wywalczyli sobie wysokie stawki.

Istnieje jednak trzecie wyjście, na razie niezbyt powszechne, ale niosące w sobie zarodki potężnych zmian cywilizacyjnych: wprowadzenie do użytku urządzeń, które jako efekt uboczny ich bezpośredniego zastosowania generują maszynowo-czytelną wersję informacji sprawozdawczej. Urządzenia takie są technologicznie całkowicie opanowane i znajdują obecnie zastosowanie w niektórych organizacjach gospodarczych, tych zwłaszcza, w których występuje konieczność prowadzenia rozległych ewidencji do celów operacyjnego zarządzania, a które nie mogą sobie już pozwolić na zatrudnienie licznego personelu księgującego lub przygotowującego na maszynowym nośniku dane dla skomputeryzowanego systemu księgowego. Dla przykładu możemy tutaj wymienić poręczne cechowniki i numeratory magazynowe, zapisujące „przy okazji" informacje na taśmie magnetycznej, kasy ewidencjonujące na maszynowo-czytelnym nośniku dokonywane transakcje i wspomniane uprzednio inteligentne urządzenia automatyzacji prac biurowych, ekstrahujące informację sprawozdawczą ze sporządzanych dokumentów.

Upowszechnienie tego rodzaju ubocznego trybu generowania informacji sprawozdawczej, zwłaszcza w połączeniu z rozwojem cyfrowej teletransmisji informacji, pozwoli objąć działalnością sprawozdawczą nieomalże wszystkie aspekty społecznej aktywności człowieka. Stworzy to warunki sprawozdawczości totalnej. Nie trzeba wielkiej wyobraźni, aby się zorientować, jakie konsekwencje cywilizacyjne przyniesie taki rozwój wypadków. Gdy każdy zakup tabliczki czekolady, a nie tylko samochodu, będzie rejestrowany i szczegółowo ewidencjonowany przy wprowadzeniu do kasy informacji o towarze i nabywcy (a to wcale nie wymaga wielkiego nakładu pracy: towar ma maszynowo-czytelną etykietkę, klient zaś płaci nie anonimowym banknotem, ale imienną kartą kredytową [19]), też, oczywiście, maszynowo-czytelną), gdy po każdym tankowaniu benzyny czy zakupie biletu kolejowego pozostaną trwałe zapisy, gdy meldunek w hotelu, datek w kościele, zakup książki, rozmowa telefoniczna (i z domu, i z automatu, gdzie płaci się kartą kredytową), każda wystawiona i zrealizowana recepta, każde wykonane badanie laboratoryjne będą zarejestrowane i szczegółowe dane o nich staną się osiągalne, wówczas nieuchronnie znikną te atrybuty cywilizacji, które określamy mianem wolności osobistej.

Można, rzecz prosta, mieć nadzieję, że tak powszechna sprawozdawczość zostanie użyta tylko dla dobra obywatela, a bezspornie ma ona takie potencjalne możliwości. Można też być przekonanym, że przetworzenie tak gigantycznej ilości informacji w celu jej wszechstronnego skorelowania będzie fizycznie niemożliwe. Zachodzi jednak obawa, że wyrywkowe przeszukanie zgromadzonych danych ewidencyjnych, np. dla zebrania kompletu wiadomości o wskazanej osobie (a więc proces o koszcie liniowym, nie kombinatorycznym!), będzie i technicznie możliwe, i uznane za uzasadnione — początkowo zapewne tylko po uzyskaniu sankcji organów sprawiedliwości, ale niestety z czasem także dla „dobra społecznego" — będzie podejmowane automatycznie przy spełnianiu pewnego ustalonego warunku jednostkowego (np. przy zakupie dwu kompletów luksusowej bielizny damskiej o różnych rozmiarach przez tego samego mężczyznę zostaje przekroczony warunek cnotliwości zachowania społecznego i kompletne dossier winowajcy automatycznie trafia do stróża moralności; a pamiętajmy, że jest do dossier bardzo kompletne!).

Nie czuję się powołany do oceny, czy taki stan rzeczy będzie społecznie dobry czy zły, tym bardziej, że ocena ta wymaga osądu etycznego, moralnego i prawnego, który możemy formułować tylko w odniesieniu do konkretnej cywilizacji. Jeśli jednak nastąpi sprawozdawczość totalna, będzie to oznaczało tak głębokie zmiany cywilizacyjne, że zmieni się zapewne poczucie tego, co dobre, a co złe. Nie są jednak

---

[18]) Gdy w kilku amerykańskich supermarketach pojawiły się pierwsze, wysoce sprawne kasy komputerowe, odczytujące ceny z prążkowo-kodowanych nalepek na towarach, protesty kupujących doprowadziły do tego, że nie tylko cena towaru w postaci czytelnej dla człowieka musi być uwidoczniona na każdym opakowaniu (co w końcu jest zrozumiałe), ale także rachunek wystawiony klientowi przez kasę komputerową musi zawierać, oprócz sumy i ceny poszczególnych towarów, szczegółowe zestawienie nazw, cen jednostkowych i cen cząstkowych (tj. poszczególne pozycje rachunku wyglądają mniej więcej tak: 2 puszki sardynek po 3½ uncji każda, po 47 centów za sztukę = 94 centy). NB, konieczność sprostania tym niemałym obowiązkom podniosła cenę kasy do ok. 20 tys. dol. (czyli do ok. 150 tys. dol. za wyposażenie typowego supermarketu o 7 stanowiskach kasowych), co nieco ochłodziło zainteresowanie handlowców (supermarkety mają bardzo niską marżę, rzędu 2%), wskutek czego na razie tylko ok. 400 na 33 tys. supermarketów w USA ma takie wyposażenie, a prognoza wzrostu do 1980 r. wymienia liczbę 1000 supermarketów o całkowicie skomputeryzowanych kasach.

(Według tygodnika „Time" z 16 października 1978 r.)

[19]) Rozpowszechnienie kart kredytowych nie bardzo pozwala na wybór „płacenia" gotówką czy kartą kredytową. Osoba płacąca gotówką sprawia więcej kłopotu i niewygód (bo przyjętą gotówką trzeba się opiekować dużo staranniej niż informacją, która dla osób trzecich nie ma wartości handlowej), a także jest podejrzana (facet ma coś do ukrycia albo, co gorsza, odmówiono mu prawa posługiwania się kartą kredytową). Już dziś trudno wypożyczyć samochód czy nawet wynająć pokój w hotelu, gdy się oświadczy, że należność będzie regulowana gotówką (czekiem podróżniczym), mimo że obie te usługi jeszcze kilka lat temu bez trudu można było opłacać gotówką

zupełnie pozbawione sensu głosy ostrzegające, że o ile dotychczasowe ciągoty do sprawozdawczości totalnej natrafiały na potężne bariery w postaci technicznej niemocy i dezintegracji organizacyjnej, o tyle — ponieważ informatyka usuwa te trudności — w nadchodzących latach dążeniom ku niepohamowanemu wzrostowi sprawozdawczości należy przeciwdziałać świadomie, nie licząc tylko na przeciwdziałanie „sił natury".

Komputeryzacja czynności finansowo-księgowych sama w sobie jest zjawiskiem czysto ekonomicznym, tym powszechniejszym, im silniejsze są odpowiednie bodźce. Bank nie korzystający ze zdobyczy informatyki traci klientów, którzy nie chcą tracić czasu na stanie w kolejce ani odsetek od sum zbyt długo przebywających w transferze. Bank stosujący metody i środki informatyczne woli otrzymywać od instytucjonalnych klientów magnetyczne nośniki zapisów zleceń wypłat niż czeki, które trzeba ręcznie przepisywać na karty dziurkowane czy — na koszt banku! — nanosić na taśmę magnetyczną. Wzrastający udział bezgotówkowego i bezczekowego sposobu dokonywania transakcji finansowych (jeszcze raz karty kredytowej!) zmusza do daleko posuniętej komputeryzacji obsługi tych transakcji.

Zautomatyzowanie obrotów finansowych w bankach wywiera silny nacisk na stosowanie podobnych metod i środków w innych sferach aktywności społecznej. Rozliczenia za dobra materialne i usługi, kredytowanie nie tylko bankowe, ale i wzajemne, itp. — wszystkie te aspekty życia gospodarczego muszą być dostosowane do form dyktowanych przez banki (warto przy tym pamiętać, że banki bodaj jako pierwsze z wielkich instytucji cywilizacji przemysłowej bez ograniczeń poddały się urokom komputeryzacji). Dodajmy do tego powszechność operacji giełdowych w kapitalistycznym wariancie cywilizacji przemysłowej, spowodowaną popieranym ze względów polityczno-ideowych i ekonomicznych rozszerzeniem kręgu osób posiadających akcje i inne papiery wartościowe, a łatwo zrozumiemy, dlaczego ta sfera zastosowań informatyki w zarządzaniu tak bardzo wyprzedziła pozostałe i dlaczego ona właśnie wywiera przemożny wpływ na dalszą informatyzację, tym razem już sieciową.

Opatrzony trzyliterowym skrótem EFT — *electronic fund transfer* — globalny system przepływu pieniędzy przy zupełnej eliminacji znaku pieniężnego i czeku, upragniony cel wielkich banków i (trzeba to dodać) szalenie wygodna forma regulowania należności obywateli, jest potężnym sprzymierzeńcem tendencji do totalnej sprawozdawczości. To właśnie przy okazji EFT zostanie wdrożona podstawowa baza techniczna umożliwiająca jej realizację. A jednocześnie nasuwa się refleksja o niemożliwości przeciwstawienia się rozwojowi EFT w jednym kraju czy grupie krajów, bez wyraźnego pogłębienia wzajemnej izolacji. Współpraca, handel i turystyka wymagają zbyt wielu i zbyt częstych operacji finansowych, aby system rozliczeń mógł za nimi nadążać przy EFT obejmującym tylko część krajów rozwiniętych.

Już dziś banki akceptujące płatności czekowe i realizujące czeki podróżnicze, a nie dysponujące elektroniczną łącznością między swoim systemem przetwarzania informacji i komputerami banków, na które te czeki zostały wystawione, ponoszą znaczne straty z tytułu opóźnień w inkasie należności (pokrycia czeków). Ale jeśli dzisiaj jeszcze te banki mogą się pocieszać, że średnio tyleż tracą, co zyskują dzięki opóźnieniom realizacji „własnych" czeków, to nie można tego powiedzieć o organizacjach handlowych i usługowych nie emitujących własnych papierów kredytowych. („Orbis" ponosi wcale pokaźne straty z tytułu kilkumiesięcznych opóźnień między przyjęciem czeku podróżniczego od klienta a wpłynięciem do kasy równoważnej kwoty).

Można przypuszczać, że w miarę jak „informatyczna", bezgotówkowa działalność finansowo-księgowa będzie się stawała normą obowiązującą de facto w rozliczeniach międzynarodowych, zostaną sformułowane odpowiednie przepisy prawne, wymagające prowadzenia „ksiąg informatycznych" w taki sposób, aby umożliwić sprawowanie kontroli finansowej również w sposób informatyczny. Nie byłby to pierwszy przypadek, gdy wymagania współpracy międzynarodowej wymuszają przyjęcie rozwiązań, których poszczególne strony wcale nie uznają za potrzebne stosować na własny użytek [20]).

[20]) Przypomina to następujące, wielce pouczające wydarzenie: w 1932 r. u wybrzeży USA, podczas szalejącego sztormu holownik stracił ciągnione barki łącznie z cennym ładunkiem. Po długim procesie zapadł wyrok zwalniający towarzystwo ubezpieczeniowe od obowiązku wypłaty odszkodowania, gdyż udowodniono, że gdyby szyper usłyszał ostrzeżenie sztormowe, mógłby wejść do portu i uratować ładunek. Szyper zaś nie usłyszał ostrzeżenia — bo nie miał radia. Od tego czasu nikt już nie uważa wyposażenia ubezpieczanej jednostki pływającej w radio za coś, o czym decyduje opinia właściciela (por. R. Bigelow, Callaghan Pub., 1977, 8, s. 149)

Wpływ informatycznych form prowadzenia działalności księgowo-finansowej na warunki cywilizacyjne społeczeństwa jest poważny; na szczególną uwagę zasługują dwa czynniki. Po pierwsze, przeciętny obywatel najwcześniej styka się z informatyką właśnie w tej sferze, jest ona dla wielu ludzi reprezentantem tego nowego, co wnosi informatyka, a przez to w bardzo znacznym stopniu wpływa na kształtowanie opinii publicznej i postaw względem całej informatyki. Po drugie, dzięki zastosowaniu komputerów do automatyzacji operacji finansowo-księgowych nastąpiła najpowszechniejsza ze wszystkich dotychczasowych zmian cywilizacyjnych wywołanych informatyzacją: odejście od takich form rozliczeń, jakie obowiązywały od stuleci. Raz jeszcze wspomnimy o kartach kredytowych, o „plastykowym pieniądzu". Oprócz skutków czysto ekonomicznych, o których już mówiliśmy, obok potencjalnych możliwości, jakie mają względem totalnej sprawozdawczości, o czym już też mowiliśmy, wspomnimy jeszcze o konsekwencjach etyczno-moralnych.

Klasyczny etos cywilizacji przemysłowej, ukształtowany przez rewolucje burżuazyjne (i w tym szczególnym aspekcie wzmocniony jeszcze przez rewolucje socjalistyczne), surowy etos purytanów i kalwinistów uznawał za niemoralne wydawanie pieniędzy na cele spożycia ponad to, co niezbędne dla regeneracji siły roboczej. Wszelkie nadwyżki miały być kapitałem, pracującym dla właściciela, albo przynajmniej oszczędzanym na czarną godzinę lub na starość. Zaciąganie pożyczki było uważane za moralnie co najmniej dwuznaczne, usprawiedliwione tylko w celu założenia lub rozwoju interesu, na zakup domu (stanowiącego naturalne zabezpieczenie kredytu) lub w sytuacjach rzeczywiście krytycznych. Karta kredytowa zachęca do wydawania ponad stan konta, byleby tylko w końcu okresu obrachunkowego znalazło się pokrycie, a ponieważ nadwyżki kapitału bankowego szukają klienta, nietrudno ewentualne debity zamienić na oprocentowane pożyczki. W obowiązującym kodeksie moralno-etycznym pojawiają się akcenty hedonistyczne; dążenie do użycia, rozbudzone postawy konsumpcyjne, potrzebne do utrzymania koniunktury w bogatych cywilizacjach przemysłowych, znajdują w karcie kredytowej swój wyraz, symbol i instrument.

Automatyczne przetwarzanie danych w celach operacyjno-decyzyjnych ma być dziedziną, w której — wedle jej apologetów — informatyka sprawdza się najpełniej jako narzędzie doskonałe: pozwala dobremu managerowi pracować jeszcze lepiej, a złemu — pomaga lepiej zorganizować produkcję, zbyt i zaopatrzenie. Co gorliwsi entuzjaści komputerowego zarządzania wysuwają nawet tezę, iż optymalne zarządzanie gospodarcze jest tak nierozerwalnie związane z komputerami, że jedno równa się drugiemu.

Oddzieliwszy od rozległej dziedziny zarządzania czynności sprawozdawcze i finansowo-księgowe (co jest, powtarzam, znacznym uproszczeniem), wypreparowaliśmy obiekt naszych obecnych zainteresowań: bieżące, operacyjne zarządzanie i planowanie gospodarcze, polegające na ciągłym podejmowaniu decyzji. O tym, że decyzja jest tym trafniejsza, im większą ilość relewantnych wiadomości poddaje się obiektywnej analizie przy jej podejmowaniu, wie każdy słuchacz nawet najbardziej skróconych kursów zarządzania. Jest rzeczą równie oczywistą, że jeśli ilość informacji, którą należy przetworzyć dla uzyskania relewantnych wiadomości, jest duża, a czas przeznaczony na podejmowanie decyzji — niewielki, techniki informatyczne mogą być (i są) niezmiernie użyteczne, pod warunkiem znajomości i poprawnego zalgorytmizowania procesu obiektywnej analizy danych.

Nieco rzadziej natomiast porusza się drugą stronę zagadnienia podejmowania efektywnych decyzji: aby można było je podejmować, przestrzeń decyzji dopuszczalnych powinna być dostatecznie rozległa, obejmować rozmaitość decyzji dobrych. Jeśli na swobodę decydenta nałożymy zbyt wiele ograniczeń, w pozostawionej mu przestrzeni mogą znaleźć się same złe decyzje, a najmniej zła ze złych nie jest jeszcze decyzją dobrą.

Nie jest to tylko niezbyt zgrabny aforyzm: kryteria optymalności decyzji są zazwyczaj formułowane w kategoriach, które odnoszą się do zbioru decyzji dobrych, służą do wyselekcjonowania najlepszych z dobrych. Te same kryteria zastosowane do zbioru decyzji złych mogą okazać się w ogóle bez sensu. Jeśli ograniczymy swobodę podejmowania decyzji, a zwłaszcza jeśli w kolejno następujących okresach będziemy ją ograniczali w coraz to inny sposób, lokalnie (w sensie czasowym) najlepsze decyzje utworzą łańcuch, który może bardzo znacznie różnić się od ciągu decyzyjnego, jaki byłby optymalny nawet przy tych samych ograniczeniach lokalnych, ale uwzględnianych globalnie.

Komputerowa realizacja algorytmów podejmowania decyzji, przy jednoczesnym arbitralnym (z punktu widzenia rozważanego układu gospodarczego) nakładaniu ograniczeń na przestrzeń decyzji dopuszczalnych, nie tylko niekoniecznie prowadzi do optymalnego zarządzania, ale może być wręcz szkodliwa ze względu na pozorną obiektywizację „obliczanych" decyzji.

Ponadto, podejmowanie decyzji przy użyciu metod i środków informatycznych nie jest ani trochę mniej wrażliwe na błędy informacji dostarczanej decydentowi niż decydowanie tradycyjne, a środki informatyczne nie mają w sobie nic wrodzonego, co by zapobiegało omyłkom czy oszustwom (aczkolwiek te ostatnie mogą przyjmować nieco inną postać, co jednak nie zmienia istoty sprawy). Owszem, użycie środków informatycznych pozwala stosować na większą skalę działania kontrolne, z drugiej jednak strony informacje są wtedy rzadziej przeglądane przez ludzi i różne absurdalne dane i niewyjaśnione, dziwne „zbiegi okoliczności", niejako automatycznie wzbudzające podejrzliwość człowieka, przechodzą gładko przez komputerowy system przetwarzania informacji, nie zaprogramowany do wykrywania konkretnie wyspecyfikowanych „dziwności" (a inne niż te wyspecyfikowane i tak „przejdą"!).

Na tę słabość komputerowych systemów zarządzania zwracano uwagę już przed laty [21]), ostrzegano też przed nadmiernym optymizmem w prognozowaniu sukcesów informatycznego zarządzania na skalę tak wielką, do jakiej rzekomo predestynują warunki społeczno-ekonomiczne ustroju socjalistycznego, wskazując na nieznane własności obliczeniowe odpowiednio wielkich układów zależności algebraicznych między danymi obarczonymi błędami.

Ostrzeżenia te zostały prawie całkowicie pominięte w krótkim, na szczęście, okresie publicystycznej euforii, kiedy to informatyce przypisywano rolę panaceum na trudności zarządzania i nawet na trudności gospodarcze. Okres ten przyniósł wielorakie szkody nie tylko samej informatyce, która w dużym stopniu straciła kredyt zaufania społecznego i, nie spełniwszy obietnic entuzjastów, zawiodła tych działaczy politycznych, którzy w nie uwierzyli, ale także — choć w mniejszym stopniu — samej gospodarce (na skutek inwestowania w systemy, które delikatnie powiedziawszy „nie sprawdziły się").

Wielokrotnie powtarzano — a i dziś się powtarza, aczkolwiek mniej publicznie — że skoro managerowie wielkich przedsiębiorstw kapitalistycznych w dużym stopniu wspomagają się w swej działalności systemami informatycznymi, to i w naszych warunkach systemy takie przyniosą podobne korzyści. Niestety, analogie tego rodzaju są nader powierzchowne i często po prostu fałszywe; posługiwanie się nimi świadczy o przedkładaniu wiary nad rozsądek, o braku umiejętności stosowania elementarnych zasad ekonomiki.

Przetwarzanie informacji, nawet przy założeniu, że nie jest ona obciążona błędem, nie może zastąpić podstawowych mechanizmów ekonomicznych: ze świadomości braku surowca czy półproduktu nie zrodzi się jego dostawa, jeśli danego towaru brak na dostępnym rynku. W warunkach częstych niedoborów wielu towarów, tylko pozornie można łagodzić powstające trudności lepszym przetwarzaniem informacji. Zaopatrzeniowiec fabryki będzie kupował potrzebne do produkcji materiały nie wtedy i nie w takich ilościach, jak to wynika z obiektywnie optymalnego harmonogramu, lecz wtedy, kiedy może, i w takich ilościach, jak może. W konsekwencji będzie on zainteresowany w zakamuflowaniu tego faktu, aby informacje przezeń dostarczane ukrywały stan rzeczy, o którym wie, że jest niewłaściwy z punktu widzenia obiektywnych formuł ekonomicznych (i w pełni zasługuje na potępienie ze strony władz zwierzchnich), ale który jest najlepszą taktyką lokalną.

Rozbudowa kontroli i reglamentacji — czynności informacjochłonnych — wzmaga dążność organów zarządzania gospodarczego szczebla wyższego niż bezpośrednio produkcyjny do stosowania systemów informatycznych, ale jednocześnie — na skutek stosowania zasady wielopoziomowej ingerencji w proste transakcje kupna/sprzedaży — wydłuża czas upływający między powstaniem zapotrzebowania a jego spełnieniem do wymiarów przekraczających horyzont czasowy optymalnego zarządzania lokalnego, tym samym znacznie ograniczając skuteczność stosowania informacjochłonnych metod zarządzania na szczeblu bezpośrednio produkcyjnym. Z drugiej zaś strony, nieuchronnie związany z reglamentacją mechanizm interwencji, z natury rzeczy nieformalny i nieformalizowalny, powoduje dalsze zmniejszenie praktycznej wartości systemów informatycznych.

Od czasu do czasu pojawiają się głosy (aczkolwiek, na szczęście, coraz rzadziej), twierdzące, że dokładne i szczegółowe planowanie może przeciwdziałać tego rodzaju konfliktom, a nawet może doprowadzić do stabilnych, poprawnych rozwiązań. Pominąwszy (wątpliwą — przynajmniej dla mnie) możliwość uzyskania rozwiązań tak ogromnych i bardzo złożonych układów zależności w sposób matematycznie rzetelny, akceptując tego rodzaju tezę musielibyśmy stanąć na stanowisku całkowicie idealistycznym, gdyż zgodność bogatej i dynamicznie zmieniającej się rzeczywistości z założeniami statycznego planu — jeśli nawet w pewnej chwili byłaby zupełna — na skutek działań przypadku (sił przyrody, kontaktów z mniej planowanym życiem gospodarczym w innych krajach) wnet przestanie obowiązywać. A im dokładniej został ułożony plan, im precyzyjniej go wyliczono, im silniejsze są więzy i pośrednictwa między składowymi procesu gospodarczego, tym większe i rozleglejsze perturbacje wynikają z wprowadzania każdej korekty.

Zjawisko to, dobrze znane projektantom i konstruktorom systemów operacyjnych, nauczyło tych specjalistów nie polegać na zbyt precyzyjnym koordynowaniu współpracujących procesów i strategię zewnętrznej koordynacji zastępować zasadą współdziałania autonomicznych procesów. Niestety, aby ta ostatnia mogła być realizowana, niezbędne jest przyjęcie i konsekwentne przestrzeganie zasady słabego pośrednictwa, ograniczającej przepływ informacji między składowymi i wprowadzającej programowe „luzy" na styku między nimi. Być może upowszechnienie tych wyników będzie największą przysługą, jaką informatyka okaże praktyce zarządzania ekonomicznego.

Aby zarządzanie było skuteczne, jego mechanizmy muszą być wolne od wieloznaczności, a zwłaszcza świadomego fałszowania informacji. Wieloszczeblowa struktura zarządzania, ingerująca w imię globalnych interesów społecznych w bieżące zarządzanie szczebla produkcyjnego, nie służy osiągnięciu tego celu. Ponieważ wiadomo, że szczeble wyższe „obetną" planowane zapotrzebowanie na surowce i półprodukty, etaty, środki dewizowe i in. już w chwili planowania zawyża się je, stosując „współczynniki bezpieczeństwa", co prowadzi do nader niebezpiecznego traktowania instrumentów planowania i zarządzania w sposób dualny: z jednej strony — dane „papierowe", z drugiej — rzeczywiste. Jeśli w takich warunkach wprowadzimy system informatyczny zarządzania, zwłaszcza połączony ze sprawozdawczością przeznaczoną dla wyższych szczebli zarządzania, jest rzeczą psychologicznie naturalną, że będzie on „karmiony" danymi papierowymi, a rzeczywiste zarządzanie będzie posługiwać się innymi danymi. W rezultacie, obok skomputeryzowanego zarządzania funkcjonuje drugie, nie sformalizowane. W tym m.in. leży przyczyna tego, że wprowadzone informatyczne systemy zarządzania nie spowodowały w Polsce zmniejszenia zatrudnienia w administracji gospodarczej przedsiębiorstw, zjednoczeń i innych jednostek. Ktoś przecież musi realizować to nieformalne zarządzanie!

Nietrudno widzieć, że skuteczność informatycznych metod zarządzania jest poważnie ograniczona stosowaniem restrykcji w mechanizmach finansowych. W układzie, w którym pieniądz utracił swoje pierwotne znaczenie, nie jest uniwersalną miarą wartości towaru i nie może być swobodnie użyty do jego zakupu — a taką przecież sytuację mamy w Polsce [22]) — ogromnie trudne (jeśli w ogóle możliwe!) staje się ustalenie obiektywnych, obliczalnych kryteriów optymalizacji decyzji.

---

[21]) Por. np. pracę niniejszego autora „Maszyny matematyczne a społeczeństwo", wydrukowaną w zbiorze „Naukowe Problemy Maszyn Matematycznych", Warszawa, 1970, PWN, s. 131—135

[22]) W praktyce życia gospodarczego w Polsce występuje tyle limitów, że mamy do czynienia z „pieniądzem złożonym". Złotówka ma pewną liczbę groszy inwestycyjnych, pewną liczbę groszy funduszu płac, pewną liczbę groszy dewizowych, itd. Kłopot polega na tym, że kompozycja złotówki złożonej zmienia się w czasie, a także w zależności od tego, kto nią dysponuje. Nagłe zamrożenia inwestycji, limitów importowych, itp. gwałtownie zmieniają skład złotówki złożonej, podobnie jak np. przyznawane konkretnemu przedsiębiorstwu wolnodewizowe kredyty Banku Handlowego wzmacniają te złotówki złożone, którymi operuje dane przedsiębiorstwo. Nie dziwota więc, że przedsiębiorstwa nie mogą ani na prawdę rozliczać się między sobą (gdyż operują różnymi złotówkami złożonymi), ani planować (a więc i zarządzać) na podstawie jednolitych kryteriów finansowych (gdyż nawet w obrębie jednej jednostki kompozycja złotówki złożonej ulega zmianie w okresie objętym planami)

W warunkach socjalistycznej cywilizacji przemysłowej, ze względów głęboko humanistycznych odrzucono podstawowe prawo gospodarki kapitalistycznej, które w uproszczeniu można wyrazić zwrotem „wszystko do kupienia". Marksowe równania o surdutach i kwarterach pszenicy nie obowiązują w praktyce społeczeństw socjalistycznych, a w każdym razie nie obowiązują w obrotach pozarynkowych [23]). W rezultacie techniki zarządzania dobrze dostosowane do mechanizmów gospodarki kapitalistycznej — a one to właśnie umożliwiły szerokie zastosowanie informatyki do zarządzania w innych krajach — okazują się nieadekwatne w społeczno-ekonomicznych warunkach krajów socjalistycznych.

Nie znaczy to oczywiście, że istnieje jakaś zasadnicza niemożność stosowania metod i środków informatyki do zarządzania w warunkach gospodarki socjalistycznej. Dwa zasadne wnioski, jakie można wyciągnąć z powyższych rozważań, brzmią zupełnie inaczej:

1. Nie należy spodziewać się, by samo zastosowanie informatyki usprawniło zarządzanie, czy to przez szybsze, obszerniejsze i bardziej wielostronne przetwarzanie informacji, czy to przez usprawnienia organizacyjne, jakie, istotnie, może ona wymusić na wprowadzającym ją przedsiębiorstwie czy instytucji.

2. Efektywne zastosowanie metod i środków informatyki w zarządzaniu (podobnie jak i w każdej innej dziedzinie) wymaga, aby istniały ścisłe i poprawne algorytmy rozwiązywania istotnych problemów zarządzania. To zaś implikuje ścisłe, a więc matematyczne (tj. nie uciekające się do gestykulacji, pomocy wizualnych ani oracji) sformułowanie problemów, jakie ma rozwiązywać zarządzanie w tych funkcjach, które mają podlegać komputeryzacji.

Jeśli przez chwilę choćby zastanowimy się nad tymi wnioskami, a zwłaszcza nad drugim, mającym znacznie konstruktywniejszy charakter, zrozumiemy, dlaczego dużo częściej możemy napotkać (nie tylko w Polsce czy innych krajach socjalistycznych) udane, efektywne i społecznie zaakceptowane systemy przetwarzania informacji wykonujące czynności cząstkowe, o dobrze określonych celach i algorytmach: sporządzające listy płac, saldujące konta bankowe, adresujące listy, sporządzające wykazy i zestawienia, a więc realizujące tzw. głupie czynności, niż bardziej ambitne w sferze życzeń, ale mniej ściśle zdefiniowane czynności całościowe, takie jak np. układanie optymalnych planów gospodarczych. I w tej zresztą ostatniej dziedzinie częściej można spotkać udane systemy rozpisujące np. plan tygodniowy na dzienne, a te — na zadania dla poszczególnych oddziałów, czyli systemy analityczne, niż systemy ustalania planu z bilansu możliwości i wymagań zewnętrznych, czyli systemy syntetyczne.

I nie trzeba się na informatykę gniewać, że tak właśnie jest; nawet gdy czujemy się zawiedzeni, że nie jest tak, jak obiecywali nam prorocy, którzy w rozwoju informatyki upatrywali przyczynę, a nie — co bliższe rzeczywistości — skutek rozwoju gospodarczego, następującego w obecności kilku jeszcze dodatkowych czynników [24]).

Wiele z tego, co dotychczas powiedziałem, miało jeden wspólny cel. Pragnąłem mianowicie rozwiać cień złudzeń, które być może ktoś żywi — złudzeń, że mała liczba zainstalowanych w Polsce komputerów oraz mierne na ogół skutki ekonomiczne ich zastosowania wynikają ze szczególnie małych zdolności intelektualnych osób odpowiedzialnych za informatykę, z wad narodowych, z ciągu pomyłek czy innych, podobnych przyczyn subiektywnych. Innymi słowy, złudzeń, że gdyby tylko inaczej pokierować rozwojem informatyki w Polsce, to byłoby z nią znacznie lepiej. Być może zamiast tych wszystkich rozważań, które przytoczyłem, wystarczyłoby odwołać się do prostego stwierdzenia, że właśnie gdy postulujemy istotne znaczenie informatyki w całej strukturze stosunków gospodarczo-społecznych, jej losy nie mogą zależeć w decydującej mierze od czynników subiektywnych, lecz muszą być — w podstawowym zarysie — determinowane czynnikami obiektywnymi, głównie rozwojem ekonomicznym społeczeństwa. Obawiałem się jednak, że tak suche stwierdzenie powyższej prawdy może być przyjęte jako wygłoszenie sloganu i że nie przekona mych czytelników.

---

[23]) O ile stosunkowo niewiele towarów rynkowych nabywa się nie za pieniądze, lecz za kombinację pieniędzy i talonów, o tyle w obrotach pozarynkowych, a zwłaszcza w działalności przemysłu, obowiązują liczne i często zmieniające się limity, częstokroć znacznie ważniejsze niż stan konta bankowego. Przy braku odpowiednika rynku dla limitów, które nie podlegają prawom wymiany towarowej, mamy do czynienia nie tylko z utrudnieniem stosowania jednolitych kryteriów finansowych w zarządzaniu (por. poprzedni przypis), ale także z zanegowaniem podstawowych mechanizmów obrotu dóbr materialnych, znanych gospodarce kapitalistycznej

[24]) Trudno oprzeć się chęci strawestowania cytowanego już hasła „informatyka kluczem do dobrobytu" na hasło „dobrobyt kluczem do informatyki". Zauważmy, że obydwa te hasła zgadzają się z zaobserwowaną korelacją między wysokością dochodu narodowego przypadającego na głowę ludności i liczbą komputerów przypadającą na milion zatrudnionych poza rolnictwem: o ile jednak pierwsze hasło sugeruje, że to informatyka wytwarza stan zamożności społecznej, o tyle drugie — eksponuje fakt, że odpowiedni poziom zamożności społecznej stwarza dobry grunt dla rozwoju informatyki. Mimo, że i drugie hasło jest fałszywe (exemplum: Arabia Saudyjska), gdyż nie uwzględnia innych warunków rozwoju informatyki, np. poziomu oświaty, kultury i obycia cywilizacyjnego społeczeństwa, a także uwarunkowań moralno-etycznych, jest ono znacznie bliższe prawdy. Przy okazji, chciałbym poświęcić kilka słów innej obiegowej nieprawdzie, krążącej wśród publicystów i działaczy gospodarczych, że to niby dla nowoczesności państwa niezbędne jest produkowanie komputerów. Zwolennicy tej teorii nie znajdują nic dziwnego w kontrprzykładach Szwajcarii, Kanady, Włoch, Belgii, Norwegii czy Australii, które najspokojniej pod słońcem rozwijają się zupełnie nieźle, nie produkując komputerów na większą skalę, ani nie są przerażeni wysokością rządowych subsydiów, jakie Francja, Wielka Brytania i — w znacznym stopniu RFN — bezpowrotnie utopiły w podtrzymanie swoich przemysłów komputerowych. Naturalnie, z nieprodukowania komputerów wcale nie wynika ich niestosowanie, wszystkie wyliczone kraje stosują ich bardzo wiele!

Ciąg dalszy w następnym numerze

CONVENTION INFORMATIQUE 80

Międzynarodowa konferencja CONVENTION INFORMATIQUE, organizowana corocznie w powiązaniu z paryską wystawą sprzętu informatycznego, biurowego i komunikacyjnego SICOB odbędzie się w roku bieżącym w dniach od 15 do 19 września. Podobnie jak w latach poprzednich miejscem obrad będzie paryski Palais de Congres, który może pomieścić spodziewaną przez organizatorów liczbę ok. 2500 uczestników z ok. 40 krajów.

Hasłem przewodnim tegorocznej, jedenastej już, konferencji będzie „Informatyka i informacja", akcentujące decydującą rolę metod i narzędzi informatycznych w procesach usprawnienia obiegu informacji. Komitet programowy konferencji ustalił następującą ramową tematykę obrad:

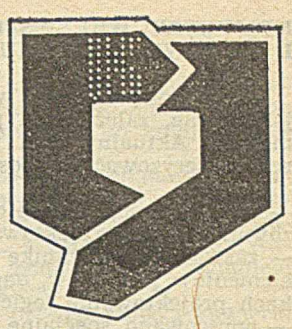
- **komputeryzacja dla usprawnienia informacji**
- **analiza danych**
- **przygotowanie danych**
- **oprogramowanie**
- **sprzęt komputerowy**
- **bazy danych**
- **teleprzetwarzanie**
- **kontrola i ocena wydajności**
- **nowe metody i systemy oraz gotowe rozwiązania**
- **analiza ekonomiczna zastosowań**
- **szkolenie.**

Organizatorzy zapraszają do zgłaszania w terminie do 29.2.80 tematów referatów wraz z krótkim streszczeniem (do 25 wierszy) oraz życiorysem autora w języku angielskim lub francuskim. Decyzje przyjęcia referatów zostaną przekazane autorom w ciągu maja z zobowiązaniem nadesłania pełnego tekstu do końca czerwca br.

Zainteresowani mogą przesyłać zgłoszenia i otrzymać dodatkowe informacje pod adresem:

CONVENTION INFORMATIQUE
6 place de Valois, 75001 Paris (France)
tel. 261-52-42

MAREK GRENIEWSKI
Ośrodek Elektronicznego Przetwarzania Danych
Handlu Wewnętrznego i Usług
Warszawa



Współczesne metody projektowania systemów informatycznych

Ostatnie dziesięć — piętnaście lat to okres szybkiego rozwoju i różnicowania się systemów informatycznych stosowanych w dziedzinie zarządzania. Przykładowo dają się wyróżnić: systemy zbierania danych, systemy przygotowania i wstępnego przetwarzania danych, systemy przetwarzania danych, systemy prowadzenia baz danych, systemy zapytaniowe, systemy dystrybucji informacji (w szczególności systemy SIK), systemy wykonawcze. Współczesne wspomaganie komputerowe zarządzania charakteryzuje się tym, że składa się na nie szereg wzajemnie współpracujących systemów. Część z nich wykorzystuje własne (tzw. dedykowane) konfiguracje sprzętu komputerowego, część zaś wykorzystuje konfiguracje sprzętowe wspólnie z innymi systemami.

Wysokie wymagania dotyczące niezawodności oraz złożoność współczesnych systemów spowodowały, że poważnie zmieniły się metody ich projektowania, budowy i wdrażania.

NIEZAWODNOŚĆ SYSTEMOWA A METODYKA PROJEKTOWANIA

W procesie projektowania systemu należy pamiętać o wszystkich grupach elementów składających się na niezawodność systemową [1]. Elementy te dają się ująć w czterech grupach:

— odtwarzalność stanów systemów informatycznych oraz odtwarzalność działania konfiguracji sprzętowych
— rekonfiguracja sprzętu komputerowego oraz rozszerzanie lub zmniejszanie liczby i zakresu funkcji użytkowych realizowanych przez system informatyczny
— usuwanie zakleszczeń powstałych na skutek sprzecznych interesów różnych procesów realizowanych w ramach danego systemu informatycznego
— ochrona zbiorów danych przed rozprzestrzenianiem się błędów powstałych w wyniku awarii.

Aby zapewnić odpowiednią niezawodność systemu, uwzględniającą powyższe elementy, przed przystąpieniem do projektowania należy określić możliwe sytuacje awaryjne i częstość ich występowania (dotychczasowa praktyka wykazuje, że niewiele jest rodzajów awarii, które pozwalają danemu systemowi zachować właściwość niezawodności systemowej), określić funkcje wymagane przez użytkownika, jakie ma realizować dany system. Nowoczesne rozwiązania konstrukcyjne, czyli modularne konfiguracje sprzętowe, powinny również przyczynić się do uzyskania optymalnej niezawodności. Najlepszym zaś sprawdzianem, czy praca projektantów przebiega we właściwym kierunku, są badania zaprojektowanego systemu lub wyników kolejnych faz projektowania.

CYKL ŻYCIA SYSTEMU

Projektowanie, budowa, wdrażanie i eksploatacja (tzw. cykl życia systemu) systemów problemowych, takich jak CENPLAN, czy SPIS, jak również wielkich systemów obiektowych, takich jak system Kombinatu Metalurgicznego „Huta Katowice", przebiegają wieloetapowo. Podobna prawidłowość występuje w trakcie realizacji innych systemów krajowych i zagranicznych. Niemożliwe jest bowiem jednorazowe zaprojektowanie, budowa i wdrożenie wielodziedzinowego systemu. Zanim jednak scharakteryzuję poszczególne etapy wyjaśnię trzy podstawowe a niezbyt znane pojęcia.

Przez **projekt logiczny** systemu informatycznego będziemy rozumieli projekt o strukturze hierarchicznej, złożony z abstrakcyjnych obiektów, takich jak: proces, dane, zdarzenia, warunki, przedziały czasu, parametry systemowe, dostawcy danych, użytkownicy informacji oraz wzajemnych powiązań między obiektami.

Przez **projekt fizyczny** systemu informatycznego będziemy rozumieli projekt logiczny rozszerzony o następujące typy obiektów: procesor (obiekt, który realizuje proces lub procesy), zasób (coś, co jest spożywane przez system wynikowy celem wykonania swoich funkcji), jednostka (obiekt służący do mierzenia zasobów), parametr użytkowania zasobu (użyty np. do określenia ilości zasobu wykorzystywanego przez proces) wraz z powiązaniami i/lub projekt konfiguracji sprzętu komputerowego wraz z projektem konfiguracji systemu operacyjnego wykorzystywanego przez system informatyczny.

Przez **test akceptacyjny** systemu informatycznego będziemy rozumieli sformalizowaną strategię postępowania, mającą na celu sprawdzenie zgodności systemu wynikowego z wymaganiami użytkownika, projektem logicznym i projektem fizycznym systemu.

Projektowanie złożonych systemów informatycznych przebiega zwykle w etapach: wstępnego przygotowania realizacji przedsięwzięcia, realizacyjnym, dalszego rozwoju i konserwacji systemu.

Etap wstępny przygotowania realizacji przedsięwzięcia obejmuje:
— wstępną analizę i identyfikację obiektu wraz z określeniem głównych funkcji użytkowych systemu i zdefiniowaniem otoczenia systemu
— wstępne projektowanie logiczne systemu i wstępne projektowanie testu akceptacyjnego systemu
— wstępne projektowanie fizyczne systemu i uzupełnienia projektu testu akceptacyjnego
— opracowanie strategii realizacji przedsięwzięcia, listy zadań, planu realizacji i kosztorysu przedsięwzięcia.

Etap wstępny w zasadzie nie jest powtarzany, chyba że przyszły użytkownik nie zaakceptuje proponowanych rozwiązań.

Etap realizacyjny, powtarzany dla każdej realizowanej grupy funkcji użytkowych systemu, obejmuje:
— uzupełniającą analizę obiektu
— projektowanie logiczne realizacji grupy funkcji użytkowych systemu i projektowanie odpowiedniej części testu akceptacyjnego
— projektowanie fizyczne realizacji grupy funkcji użytkowych systemu oraz uzupełnienie projektu odpowiedniej części testu akceptacyjnego
— zamawianie typowych modułów, konstruowanie i uruchamianie modułów nietypowych, opracowanie odpowiedniej części testu akceptacyjnego
— kompletowanie, integrowanie i uruchamianie całościowe danej części systemu zakończone wstępnym wykonaniem testu akceptacyjnego
— instalowanie systemu
— konwersje zbiorów danych oraz wdrażanie kolejnych funkcji użytkowych systemu
— testowanie akceptacyjne systemu w zakresie założonej realizacji
— przekazanie użytkownikowi zrealizowanych części systemu.

[1] Najogólniej mówiąc „niezawodność systemowa" to zdolność przywrócenia normalnego działania systemu po awarii powodującej powstanie błędu

KOMPUTEROWE WSPOMAGANIE PROJEKTOWANIA I BUDOWY SYSTEMU

Kiedy komputerowe wspomaganie objęło również projektowanie, zależnie od podejścia stosowano najrozmaitsze metody. Aktualny stan wiedzy w tej dziedzinie daje się scharakteryzować w sposób następujący:

— wzorując się na metodzie HIPO, opracowano specjalizowane języki projektowania logicznego i fizycznego
— zastosowano technikę baz danych do przechowywania elementów projektów, napisanych w specjalizowanych językach projektowania logicznego i fizycznego
— opracowano specjalne pakiety programów, tzw. analizatory języków projektowania, współpracujące z bazami danych
— wykorzystano uniwersalne procesory redagowania danych tekstowych *(text-editor)* dla wprowadzania do komputera elementów projektu, napisanych w specjalizowanych językach projektowania logicznego i fizycznego
— opracowano specjalne pakiety programów, tzw. generatory dokumentacji, współpracujące zarówno z procesorem redagowania danych tekstowych, jak i analizatorami, służące do prowadzenia i emisji dokumentacji systemu według ustalonych standardów
— rozwinięto technikę programów typu automatycznego sporządzania schematów blokowych *(flow-charter)* do celów diagnostycznych.

Kilka dodatkowych uwag dotyczących specjalizowanych języków projektowania logicznego. Przykładem języka projektowania logicznego jest dość szeroko rozpowszechniony język PSL *(Problem Statement Language)*, opracowany przez zespół z Uniwersytetu Michigan w Ann Arbor (USA) pod kierunkiem prof. dr. D. Teichrowa. W języku PSL wyróżnianych jest kilkanaście podstawowych typów obiektów, z których każdy opisywany jest przez pewien zespół zdań języka, tworzący tzw. sekcję. W ramach sekcji obok opisu danego obiektu pokazane są powiązania danego obiektu z innymi obiektami występującymi w projekcie.

KOMPUTEROWE METODY BADANIA POPRAWNOŚCI PROJEKTÓW

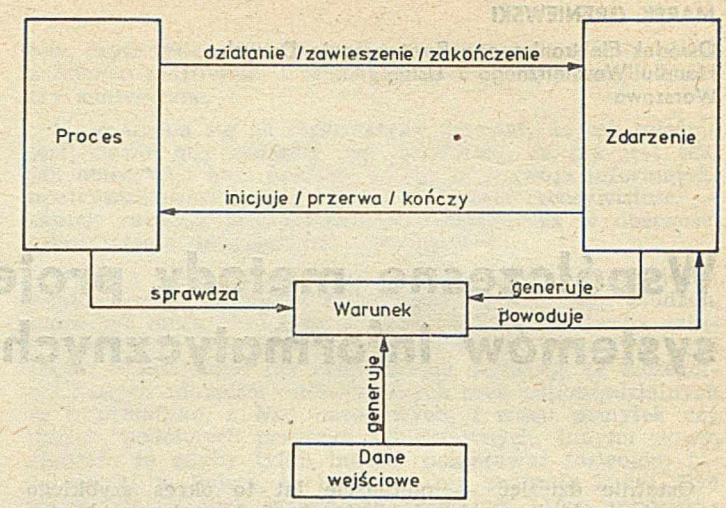
Specjalne pakiety programów (tzw. analizatory) zawierają zaprogramowane metody analizy poprawności formalnej projektu. Dotychczas stosowane i proponowane metody komputerowej analizy projektu można ująć w następujące trzy grupy:
— metody analizy Langenforsa, stanowiące rozwinięcie podejścia CODASYL przedstawionego w tzw. algebrze informacji (ang. *information algebra*)
— metody symulacyjne
— metody czasowo-zależnych sieci Petriego (tzw. *Time Petri Nets*).

Ta ostatnia grupa metod, jako najmniej znana przez naszych specjalistów, wymaga kilku dodatkowych wyjaśnień. Otóż koncepcja ta pochodzi od Ph. M. Merlina i D. J. Farbera z Uniwersytetu Kalifornia w Irvine (USA) i została po raz pierwszy opublikowana w 1975 roku. Czasowo-zależne sieci Petriego, podobnie jak klasyczne sieci Petriego, składają się z dwu rodzajów elementów, tzw. zdarzeń i warunków, połączonych zorientowanymi łukami. W odróżnieniu jednak od wersji klasycznej — z każdym zdarzeniem związane są dwie dodatnie liczby, z których pierwsza nazywana jest minimalnym czasem wystąpienia zdarzenia, liczonym od momentu wystąpienia wszystkich warunków, decydujących o wystąpieniu danego zdarzenia, druga zaś — maksymalnym czasem wystąpienia zdarzenia, liczonym podobnie jak dla czasu minimalnego.

Teoria zachowania się, czy też działania czasowo-zależnych sieci Petriego okazała się potężnym aparatem analitycznym do badania odtwarzalności i rekonfiguracji systemu.

Wyobraźmy sobie, że projekt nasz jest napisany w języku projektowania, w którym między innymi występują cztery typy obiektów: PROCES, WARUNEK, ZDARZENIE i DANE WEJŚCIOWE. Możliwe wzajemne powiązania pomiędzy obiektami tych czterech typów prezentuje rysunek obok.

Przejście od projektu do czasowo-zależnej sieci Petriego, opisującej sterowanie w naszym systemie, uzyskujemy przez ograniczenie się do obiektów typu ZDARZENIE i WARUNEK oraz ich wzajemnych powiązań. Minimalny i maksymalny czas wystąpienia ZDARZENIA należy przy tym interpretować jako parametry systemowe.



Schemat ideowy możliwych wzajemnych powiązań pomiędzy obiektami typu: proces, zdarzenie, warunek, dane wejściowe

ROZWÓJ TECHNIKI PAKIETÓW PROGRAMÓW UŻYTKOWYCH

Jedną z zasadniczych przyczyn bardzo dużej pracochłonności i długotrwałości oprogramowywania systemów informatycznych (w latach sześćdziesiątych i w pierwszej połowie lat siedemdziesiątych) było indywidualne pisanie programów. Zdaniem niektórych autorów blisko 95% programów pisanych w latach sześćdziesiątych było wielokrotnym oprogramowywaniem tych samych algorytmów z nieco innymi ograniczeniami. Większość tych programów była przy tym tak źle dokumentowana, że nie mogła być dalej modyfikowana lub rozwijana.

Szczegółowsze badania wykazały, że pracochłonność programu dobrze udokumentowanego (tzn. takiego, który może być modyfikowany lub rozwijany przez dowolnego programistę) jest 3-krotnie wyższa od pracochłonności programu przeznaczonego do stosowania tylko przez autora. Rozwiązaniem umożliwiającym szybkie i mało pracochłonne oprogramowanie zaprojektowanego systemu jest natomiast wykorzystanie tzw. generacyjnych pakietów programów użytkowych.

Generacyjny pakiet programów użytkowych składa się z następujących podstawowych części:
— interpretera parametrów generacyjnych [2]
— monitora interpretującego parametry chwilowe [3] oraz sterującego wykonaniem przebiegu realizacyjnego
— biblioteki modułów programowych, realizujących pod nadzorem monitora wykonanie poszczególnych kroków przetwarzania.

Ze względu na rodzaj realizowanych funkcji przetwarzania — generacyjne pakiety programów użytkowych można podzielić na następujące trzy klasy:
— generacyjne pakiety programów wejścia lub wyjścia
— pakiety programów zarządzania bazami danych (zwane potocznie systemami zarządzania bazami danych)
— generacyjne pakiety programów ukierunkowanych na metody (np. pakiet do obliczeń statystycznych OSIRIS).

W przypadku oprogramowywania zaprojektowanego systemu z wykorzystaniem generacyjnych pakietów programów praca projektanta-programisty sprowadza się do:
— wybrania generacyjnych pakietów programów odpowiednich dla zrealizowania oprogramowania systemu
— wygenerowania odpowiednich programów użytkowych w oparciu o wybrane generacyjne pakiety programów
— indywidualnego napisania i udokumentowania programów interfejsowych, wzajemnie łączących wygenerowane programy użytkowe.

---

[2]) Przez „parametry generacyjne" rozumie się parametry służące do nadania monitorowi oraz modułom bibliotecznym odpowiedniego kształtu dla realizacji przebiegów wchodzących w skład zaprojektowanego systemu

[3]) Przez „parametry chwilowe" rozumie się parametry służące do wariantowania realizacji przebiegu w zależności od bieżących potrzeb

Należy podkreślić, że opracowanie generacyjnego pakietu programów w przeliczeniu na jeden wiersz programu źródłowego jest ok. 9 razy bardziej pracochłonne od opracowania dobrze udokumentowanego indywidualnie pisanego programu, a tym samym ok. 27 razy bardziej pracochłonne od opracowania nieudokumentowanego programu. Wynika z tego, że jedynie dla algorytmów przeznaczonych do wielokrotnego (co najmniej 20-krotnego) wykorzystania opłaca się opracowywać generacyjne pakiety programów.

DOKUMENTOWANIE SYSTEMÓW

Istnienie złożonych systemów o długim cyklu życia nie byłoby możliwe bez jednoznacznej i odpowiednio szczegółowej dokumentacji. Można zaryzykować stwierdzenie, że umiejętność dokumentowania systemów jest tą umiejętnością, która decyduje o efektywności działania realizatorów, o sprawnym wdrażaniu i wieloletniej eksploatacji systemu. Trzeba też dodać, że bez odpowiednio opanowanej umiejętności dokumentowania, niemożliwa jest praca w dużych zespołach realizatorskich.

W krajach wysoko rozwiniętych ukształtowane zostały pewne standardy dokumentacyjne. W oparciu o te standardy można przyjąć, że dokumentacja systemu informatycznego składa się z następujących części:

— dokumentacji realizacyjnej systemu (zwanej również dokumentacją produkcyjną), obrazującej przebieg procesu powstawania systemu
— dokumentacji konstrukcyjnej, służącej do usuwania ewentualnych błędów wykrytych w toku eksploatacji oraz do wprowadzania modyfikacji i dalszego rozwoju systemu
— dokumentacji eksploatacyjnej
— dokumentacji wdrożeniowej (w szczególności dokumentacji szkoleniowej dla użytkownika)
— dokumentacji akwizycyjnej (dotyczy jedynie systemów powielarnych przeznaczonych do wielokrotnego wdrożenia).

Dokumentacja realizacyjna składa się zwykle z:

— dokumentacji bazowej realizacji przedsięwzięcia
— planu realizacji przedsięwzięcia
— dziennika realizacji przedsięwzięcia
— protokołów uzgodnień i badań
— wywiadów, ankiet, opinii itp.

Kluczową rolę odgrywa dokumentacja bazowa realizacji przedsięwzięcia, której poświęcam dalsze rozważania.

DOKUMENTACJA BAZOWA REALIZACJI PRZEDSIĘWZIĘCIA

Dokumentacja bazowa realizacji przedsięwzięcia jest tworem dynamicznym, ulegającym zmianom w miarę postępu prac nad systemem. Dlatego też powstają jej kolejne, uzupełniane redakcje.

Dokumentacja ta jest głównym środkiem komunikacji zarówno wewnątrz zespołu realizatorskiego, jak i pomiędzy zespołem realizatorskim i użytkownikiem. Z zasady prowadzi się ją w kilku egzemplarzach, a utrzymanie identyczności wszystkich egzemplarzy jest podstawowym zadaniem koordynatorów zespołu realizatorskiego.

Dokumentacja bazowa realizacji przedsięwzięcia zawiera:

— charakterystykę obiektu i jego dynamikę, opis możliwych zakłóceń oraz charakterystykę użytkownika systemu
— klasyfikację funkcji użytkowych systemu
— opis funkcji użytkowych systemu
— opis konstrukcji funkcji użytkowych systemu
— opis konfiguracji sprzętu komputerowego i jego rozmieszczenia w obiekcie
— opis bibliotek modułów programowych i procedur sterowania zadaniami programowymi systemu
— opis testu akceptacyjnego systemu
— opis strategii realizacji przedsięwzięcia
— opis organizacji przedsięwzięcia
— listy zadań składających się na przedsięwzięcie i stan ich realizacji
— opis przyjętych standardów
— opis typowego sprzętu komputerowego
— opis nietypowego (specjalnie projektowanego) sprzętu komputerowego
— opis wybranych wersji i konfiguracji systemów operacyjnych
— opis nietypowych (specjalnie projektowanych) modułów systemów operacyjnych
— opis pakietów programów wybranych dla realizacji systemu
— opis indywidualnie pisanych modułów programowych
— opis programów pomocniczych.

* * *

Powyższy szkic o współczesnych metodach projektowania systemów informatycznych, stosujących wspomaganie komputerowe, nie wyczerpuje oczywiście tematyki. Wydaje się jednak, że pozwala on wyrobić pogląd na współczesny „stan sztuki", i przyczyni się do szerszego zainteresowania problematyką wspomagania komputerowego w projektowaniu systemów informatycznych wśród projektantów i programistów.

ELŻBIETA NIEMYJSKA
Zakład Organizacji Przetwarzania Danych
Uniwersytetu Łódzkiego

Praktyczne aspekty stosowania metod strukturalnych

Koncepcja metod strukturalnych (ang. *programming productivity techniques*) w pracach projektowo-programowych oraz dokumentacyjnych zrodziła się około 10 lat temu. Już pierwsze publikacje z tej dziedziny [1]) jak również dalsze, rozwinięte prace takich prekursorów metod strukturalnych jak Dijkstra, Jackson, Warnier czy Yourdon wzbudziły szerokie zainteresowania specjalistów-informatyków ze względu na odmienność proponowanego podejścia oraz wysoką efektywność w stosunku do powszechnie wykorzystywanych metod tradycyjnych w projektowaniu i programowaniu. Metody strukturalne, według definicji E. Yourdona [5], są „...metodami logicznej analizy, projektowania, oprogramowania i dokumentacji dowolnego systemu, skomputeryzowanego lub nie ....; obejmują podejście „od góry do dołu"

[1]) Pierwsze artykuły z dziedziny metod strukturalnych, to G. Jacopini, C. Bohn: Flow Diagrams, Turning Machines and Languages with only Two Formulation Rules (1966) oraz E. Dijkstra: Structured Programming (1969)

Mgr Elżbieta Z. NIEMYJSKA ukończyła w 1977 r. Wydział Cybernetyki Ekonomicznej i Informatyki Uniwersytetu Łódzkiego. Po ukończeniu studiów pracowała w Zakładzie Organizacji Przetwarzania Danych UŁ na etacie pracownika naukowo-badawczego, a następnie naukowo-dydaktycznego. W okresie maj 1978 r. — czerwiec 1979 r. przebywała na praktyce zawodowej w Systems & Programming Division w Urzędzie Stanowym w Atlancie (USA). Brała udział w pracach grupy projektancko-programowej na potrzeby departamentu policji miasta Atlanty. Specjalizuje się w problematyce modelowania symulacyjnego i metodach strukturalnych.

(ang. *top-down*), koncepcję zespołu głównego programisty (ang. *chief programmer team*), metodę HIPO, struktury hierarchiczne, diagramy Warniera, schematy przyczynowo-skutkowe (ang. *buble structures lub cause-and-effect diagrams*) itd...".

Niezależnie od niewielkich różnic merytorycznych poszczególnych „szkół metod strukturalnych", jakie wykształciły się pod kierunkiem każdego z wymienionych twórców, odmienność proponowanego podejścia polega przede wszystkim na niesekwencyjnym, modułowym konstruowaniu hierarchicznej struktury projektu. Każdy z modułów wyodrębniany jest z punktu widzenia funkcji, jaką spełnia w ogólnej koncepcji systemu. Taka odrębność funkcyjna sprawia, że wszystkie procesy zachodzące w ramach danego modułu są jednorodne co do swej treści, np. w module obliczeniowym wykonywane są jedynie operacje arytmetyczne bądź logiczne, nie ma natomiast operacji WE/WY czy formalno-logicznej kontroli rekordów. Moduły charakteryzują się wysoką spójnością wewnętrzną (ang. *cohesion*) oraz niskim współczynnikiem powiązań międzymodułowych (ang. *coupling*). W odmienny sposób przedstawiana jest również szata graficzna koncepcji, projektu oraz dokumentacji.

Najbardziej kontrowersyjną propozycją, jaką wniosły metody strukturalne jest wyeliminowanie z języków programowania, a przynajmniej znaczne ograniczenie stosowania instrukcji skoku bezwarunkowego. G. Jacopinii i C. Bohn przeprowadzili w wyżej wspomnianym artykule dowód matematyczny potwierdzający tezę, że stosowanie instrukcji skoku bezwarunkowego w programach nie tylko nie podnosi, ale wręcz obniża efektywność ich działania ze względu na małą przejrzystość logiczną programów i stosowanie dużej liczby zwrotnic, a w rezultacie znaczne zwiększenie prawdopodobieństwa „zapętlenia się" programu w trakcie jego realizacji. Praktyka programowania wykazała jednak, że w pewnych uzasadnionych przypadkach korzystne jest stosowanie omawianej instrukcji, np. jeśli w ramach tego samego procesu testujemy jedną z wielu równolegle występujących możliwości (ang. *case structure*) lub gdy zachodzi konieczność bezwzględnego „wyjścia" z danej procedury przed końcem paragrafu, w którym ta procedura występuje.

CHARAKTERYSTYKA WYKORZYSTANIA METOD STRUKTURALNYCH

Metody strukturalne znajdują szerokie zastosowanie w przyzakładowych ośrodkach obliczeniowych w Stanach Zjednoczonych i w wielu przypadkach wyparły całkowicie stosowane dotychczas metody tradycyjne. Realizacja projektu systemu informatycznego, od fazy przedprojektowej do wdrożenia przebiega najczęściej według popularnej wśród amerykańskich praktyków-informatyków koncepcji „szkoły Yourdona".

Projekt przydzielany jest niewielkiej grupie specjalistów spełniających wszelkie wymogi zespołu głównego programisty. Etap wstępny projektowania odbywa się na zasadzie „burzy mózgów". Członkowie grupy otrzymują to samo zadanie, polegające na zaprojektowaniu sieci struktury logicznej całego systemu przy wykorzystaniu schematów przyczynowo-skutkowych. Następnie poszczególne projekty poddawane są ogólnej dyskusji, w rezultacie której wyłania się wspólna koncepcja struktury systemu. Morfologia systemu odzwierciedlona jest w postaci schematu struktury hierarchicznej z wydzieleniem funkcyjnej odrębności modułów. Struktura wewnątrzmodułowa przedstawiona zostaje w postaci diagramów Warniera.

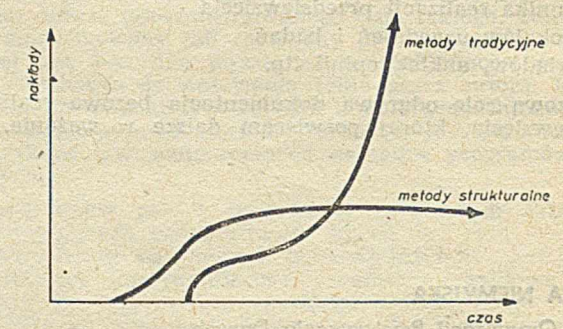
Po zatwierdzeniu całości projektu grupa podejmuje decyzje dotyczące technicznej realizacji poszczególnych modułów z punktu widzenia dostępnych urządzeń WE/WY i pojemności pamięci operacyjnej, zawartości wykorzystywanych bibliotek programów oraz wspólnej bazy danych. W przypadku dużych, skomplikowanych systemów stosowana jest technika nakładek. Poszczególne moduły kodowane i kompilowane są osobno, a następnie składowane w pamięci pomocniczej (ang. *core image library*). Kodowanie modułów odbywać się może w sposób równoległy, przy czym najbardziej doświadczeni programiści otrzymują do realizacji skomplikowane moduły kontrolne.

Natychmiast po skompilowaniu głównego modułu kontrolnego (ang. *root phase*) rozpoczyna się testowanie systemu metodą *top-down*. Polega ona na dołączaniu do każdego modułu nadrzędnego tzw. „pustych" modułów podrzędnych (ang. *overlay phases*), które wywoływane są przez moduł nadrzędny za pomocą instrukcji CALL. Rola modułów pustych polega tylko i wyłącznie na sprawdzeniu poprawności połączeń międzymodułowych oraz prawidłowości działania modułu nadrzędnego. Testowanie systemu ma charakter iteracyjny, przy czym na każdym etapie kolejne moduły puste zastępowane są modułami rzeczywistymi. W wypadku niezbyt skomplikowanych, małych systemów zamiast nakładek stosowana jest często technika segmentacji.

Dokumentacja eksploatacyjna przeznaczona dla użytkownika zaprojektowanego systemu informatycznego przedstawiana jest najczęściej w formie czytelnych i przejrzystych schematów HIPO.

KORZYŚCI WYNIKAJĄCE ZE STOSOWANIA METOD STRUKTURALNYCH

Dział marketingu „szkoły Yourdona" przeprowadził wnikliwie badania [2], w których skonfrontowano wyniki prac nad kompleksowym, złożonym projktem systemu informatycznego realizowanego metodami tradycyjnymi i strukturalnymi. Przeprowadzono porównanie, biorąc pod uwagę aspekty ekonomiczno-finansowe przedsięwzięcia, zaangażowanie emocjonalne projektantów i programistów, czas niezbędny na realizację projektu, efektywność wykorzystania komputera oraz parametry użytkowe samego systemu. Efekty badań potwierdziły w pełni wysoką efektywność ekonomiczną metod strukturalnych w porównaniu z metodami tradycyjnymi.



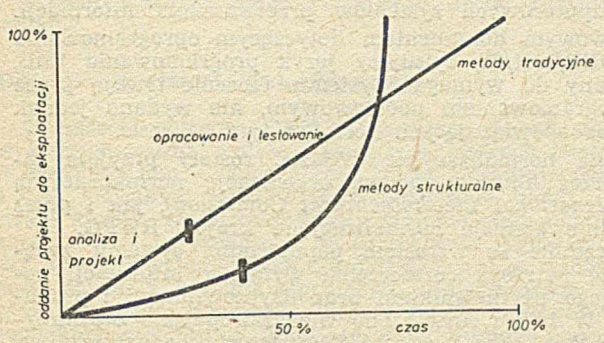
Rys. 1. Nakłady finansowe przy podejściu tradycyjnym i strukturalnym wg pracy [3]

Do głównych czynników przemawiających na korzyść tych pierwszych należało obniżenie nakładów finansowych na dany projekt (rys. 1). Wieloletnia praktyka wykazała, że koncepcja, projekt i oprogramowanie systemu przy wykorzystaniu metod tradycyjnych stanowiły około 60% ogólnych kosztów, przy czym rozkładały się one w sposób monotoniczny rosnący. Resztę czasu i kosztów pochłaniało testowanie i wdrożenie. Metody strukturalne zmieniły powyższe proporcje w sposób zasadniczy. W początkowym stadium opracowywania systemu zaobserwować można było progresywny wzrost nakładów na analizę, koncepcję i projekt. Później krzywa nakładów ustabilizowała się na pewnym określonym poziomie będącym odzwierciedleniem kosztów programowania i testowania, które to czynności prowadzone były w sposób równoległy.

[2]) Wyniki tych badań zebrane zostały w pracach:
Barry Boehm: Software and Its Impact: A Quantitative Study. DATAMATION, May 1973 oraz
Plauger P. I.: New York Times Revisited. The Yourdon Report, vol. 1, no. 3, April 1976

Niemniej ważne było zmniejszenie liczby projektantów i programistów opracowujących dany system. Przyjęcie koncepcji zespołu głównego programisty wpłynęło na lepszą alokację środków, zwiększenie odpowiedzialności zespołowej i podwyższenie wydajności pracy poszczególnych członków zespołu.

Z punktu widzenia użytkownika olbrzymią rolę odegrało skrócenie okresu oczekiwania na pozytywne wyniki testowania poszczególnych modułów i całego systemu. Zastosowanie metody *top-down*, przydzielenie skomplikowanych modułów bardziej doświadczonym programistom, ciągłe testowanie współdziałania pomiędzy modułami nadrzędnymi i podrzędnymi przy użyciu wspomnianych modułów „pustych" skróciło w znacznej mierze czas przeznaczony na testowanie poszczególnych ogniw systemu i ich współpracy oraz systemu jako całości (rys. 2).



Rys. 2. Realizacja projektu przy użyciu metod tradycyjnych i strukturalnych wg pracy [4]

Stosowanie metod strukturalnych zwiększyło elastyczność systemu i obniżyło koszty jego eksploatacji. System oparty na metodach strukturalnych charakteryzował się wysokim współczynnikiem spójności wewnętrznej [3]) i niskim współczynnikiem powiązań międzymodułowych [4]). W przypadku konieczności dokonania zmian określonego procesu wystarczyło, aby kierując się funkcją, w ramach której ten proces występował, odnaleźć odpowiedni moduł, dokonać wymaganych zmian i przetestować prawidłowość nowego rozwiązania nie naruszając struktury pozostałych modułów funkcyjnych. Jeżeli np. zachodziła konieczność dołączenia nowego procesu czy funkcji, wystarczyło skonstruować odpowiedni moduł i dołączyć go do istniejącej sieci, bez naruszenia podstawowej struktury systemu.

System zaprojektowany w oparciu o metody strukturalne charakteryzował się większą czytelnością i przejrzystością. Hierarchizacja struktury sprawiła, że funkcje, jakie system spełniał, sformułowane były w postaci odseparowanych modułów i łączyły się z innymi modułami za pomocą powiązań międzymodułowych. W ramach danej funkcji (modułu) można było wyróżnić w sposób jednoznaczny poszczególne procesy ujęte w sekcje i paragrafy. Przepływ działań odbywał się na zasadzie ciągłej interakcji pomiędzy procesami od pierwszej do ostatniej instrukcji PERFORM. Dla porównania przepływ akcji opartych na schematach blokowych z wykorzystaniem instrukcji skoku bezwarunkowego sprowadzał się do kolejnych skoków, co bardzo często prowadziło do „zapętlenia" programów.

Nie bez znaczenia był również emocjonalny stosunek projektantów i programistów do realizowanego systemu. Przejawiał się on w zwiększonym zaangażowaniu psychologicznym członków zespołu głównego programisty w sprawy projektu. Wspólne podejmowanie decyzji dotyczących kształtu poszczególnych programów i zespołowa odpowiedzialność za powodzenie każdej części projektu wpłynęła dodatnio na stosunki międzyludzkie w zespole oraz stosunek do projektowanego systemu.

Ważnym aspektem była również czytelność dokumentacji projektowo-programowej sporządzonej w oparciu o metodę HIPO.

UWAGI KOŃCOWE

Przeprowadzone badania wykazały, że zastąpienie metod tradycyjnych metodami strukturalnymi w pracach nad systemami informatycznymi jest opłacalne zarówno z punktu widzenia czynników ekonomicznych jak i pozaekonomicznych. Przy podejmowaniu decyzji dotyczącej wprowadzenia metod strukturalnych do praktyki projektowania i programowania trzeba jednak pamiętać o wysiłku, jaki należy włożyć w przeszkolenie członków grup projektancko-programowych, przygotowanie odpowiedniej bazy technicznej, a co najważniejsze, w przełamanie zakorzenionego przywiązania do metod tradycyjnych, jakie często wykazują programiści, projektanci, analitycy i przedstawiciele szczebli kierowniczych.

Dlatego też dużą rolę w rozpowszechnianiu metod strukturalnych w projektowaniu i programowaniu należy, zdaniem autorki, przypisać zajęciom z EPO prowadzonym na wyższych uczelniach oraz na kursach doskonalenia kadr kierowniczych. Studenci już na wykładach „wstęp do informatyki" powinni otrzymywać informacje dotyczące podejścia *top-down* oraz współczesnych technik projektowania i dokumentacji projektowo-programowej, a nie, jak jest to obecnie praktykowane, dopiero na ostatnich latach studiów, gdzie problemy te wykładane są jako swoista ciekawostka i z przyczyn technicznych nie mogą być wykorzystane w pracach magisterskich. Tylko wczesne zapoznanie studentów z metodami strukturalnymi pozwoli na wyrobienie praktycznych umiejętności w posługiwaniu się współczesnymi technikami projektowo-programowymi, konfrontację z metodami tradycyjnymi i w końcowym rezultacie wykształcenie wysokiej klasy projektantów i analityków systemów informatycznych na potrzeby gospodarki narodowej.

[3]) Spójność (kohezja) wg definicji Yourdona jest miarą współzależności elementów danego modułu. Yourdon [4] wyróżnia 7 typów spójności wewnętrznej modułów

[4]) Na temat powiązań międzymodułowych, a szczególnie tzw. powiązań patologicznych wypowiedział się szeroko E. Yourdon w pracy [5]

LITERATURA:

[1] Dijkstra E.: A Discipline of Programming. Englewood Cliffs, N.J. Prentice-Hall, 1974
[2] De Marco T.: Structured Analysis and System Specification. Yourdon Press, Inc. N.Y. 1978
[3] Orr K.: Structured System Development. Yourdon Press, Inc. N.Y. 1977.
[4] Yourdon E.: Techniques of Program Structure and Design. Englewood Cliffs. N.J., Prentice-Hall, 1975
[5] Yourdon E.: How to Manage Structured Programming. Yourdon Press, Inc. N.Y., 1976

Czytajcie INFORMATYKĘ

STANISŁAW KOŚLACZ, KRZYSZTOF RZYMKOWSKI
Instytut Badań Jądrowych
Świerk

CAMAC – blokowy system elektroniczny do automatyzacji pomiarów i sterowania

W latach 1950—1960 zarysowała się wyraźna tendencja do tworzenia dużych uniwersalnych systemów elektronicznych, przeznaczonych głównie do automatyzacji pomiarów w eksperymentach fizycznych lub do automatyzacji sterowania procesami produkcyjnymi. Wiele takich systemów opracowały różne organizacje międzynarodowe lub poszczególne firmy produkujące specjalizowaną aparaturę elektroniczną. Niewiele powstałych systemów ma jednak szerokie uniwersalne zastosowanie.

Przedstawiony poniżej system CAMAC *(Computer Application for Measurements and Control)* został opracowany na potrzeby atomistyki. Obecnie należy do nielicznej grupy uniwersalnych systemów elektronicznych automatyki i stosowany jest masowo w niemalże wszystkich dziedzinach nauki i techniki. Wynika to głównie z modularnej jego budowy, dzięki czemu można tworzyć różne zestawy funkcjonalne spełniające indywidualne wymagania użytkowników. Poza tym system odznacza się nowoczesnymi rozwiązaniami konstrukcyjnymi, w których uwzględniono margines umożliwiający wprowadzanie zmian i uzupełnień bez naruszania wymienności funkcjonalnej.

Produkcja światowa urządzeń CAMAC przekracza obecnie 1000 różnych typów bloków wykonawczych, sterujących, pośredniczących, umożliwiających tworzenie różnych konfiguracji układowych.

Podstawowe założenia systemu CAMAC powstały w laboratoriach fizyki jądrowej zachodnioeuropejskich ośrodków naukowych w drugiej połowie lat sześćdziesiątych. Większość tych ośrodków nie dysponowała jednak sprzętem przystosowanym do bezpośredniego połączenia z używaną aparaturą pomiarowo-kontrolną. W celu znormalizowania parametrów i właściwości systemu utworzono komitet ESONE *(European Standards On Nuclear Electronics)*, który w 1969 r. sprecyzował podstawowe parametry systemu i określił zasady działania zestawów CAMAC. Prace normalizacyjne kontynuowano w latach następnych, przy czym warto dodać, że od 1970 r. w działaniach ESONE aktywnie uczestniczyli przedstawiciele Polski.

Podstawowe postanowienia normalizacyjne są zawarte w następujących dokumentach:
EUR 4100 (PN-72/T-06530) — dotyczy zaleceń mechanicznych oraz organizacji logicznej pojedynczej kasety
EUR 4600 (PN-75/T-06532) — dotyczy zaleceń mechanicznych i organizacji logicznej wielokasetowego systemu równoległego
EUR 5100 (PN-75/T-06531) — normalizuje sygnały analogowe
EUR 6100 (PN w opracowaniu) — dotyczy zaleceń mechanicznych i organizacji logicznej wielokasetowego systemu szeregowego
EUR 6500 (PN w opracowaniu) — dotyczy zaleceń mechanicznych i struktury połączeń wielu źródeł sterowania w pojedynczej kasecie, między innymi dla jedno- i wielomikrokomputerowych systemów przetwarzania informacji.

Podstawowym dokumentem dotyczącym oprogramowania jest ESONE/IMI/01, opisujący język programowania IML, dostosowany do wymagań systemu. Obecnie trwają prace nad oprogramowaniem podstawowym, nie wydano jednak dotychczas obowiązujących norm w tym zakresie.

Zalecenia normalizacyjne CAMAC zostały przyjęte zarówno przez międzynarodową organizację normalizacyjną IEC *(International Electrotechnical Comission)*, jak i przez odpowiednie komisje normalizacyjne krajów RWPG.

Równocześnie ze wzrostem popularności systemu zwiększała się potrzeba rozszerzenia wymiany informacji pomiędzy jego użytkownikami oraz użytkownikami i producentami. W 1975 r. utworzono więc Europejskie Stowarzyszenie CAMAC *(European CAMAC Association)*, zrzeszające producentów i użytkowników systemu, m.in. polskiego producenta aparatury CAMAC — Zjednoczone Zakłady Urządzeń Jądrowych POLON.

CAMAC jest już również znany i stosowany we wszystkich krajach socjalistycznych. Potwierdzeniem wysokiej przydatności systemu do celów naukowych jest uchwała przyjęta w 1975 r. przez akademie nauk w krajach socjalistycznych, stanowiąca, że CAMAC będzie jedynym systemem używanym do automatyzacji eksperymentów naukowych.

OGÓLNE WŁAŚCIWOŚCI SYSTEMU CAMAC

System CAMAC zapewnia dwustronne połączenie i przekazywanie informacji pomiędzy układem pomiarowym a komputerem lub innym urządzeniem sterującym. Konstrukcja i logika systemu zapewniają pełną wymienność urządzeń pochodzących od różnych producentów.

Zaletą systemu jest konstrukcja blokowa, umożliwiająca modyfikację zestawów pomiarowych. Każdy blok funkcjonalny może być bezpośrednio połączony z magistralą zapewniającą przesyłanie sygnałów informacyjnych i sterujących. Konstrukcja wszystkich bloków, z wyjątkiem bloku pośredniczącego (łączącego zestaw bądź z komputerem, bądź z eksperymentem), jest niezależna od rodzaju urządzenia sterującego (komputera) i eksperymentu.

Systemy jednokasetowe mogą być łączone w systemy wielokasetowe za pomocą magistral zewnętrznych równoległej lub szeregowej, umożliwiając budowę nawet bardzo złożonych sieci sterowanych jednym lub kilkoma komputerami.

Dr Stanisław KOŚLACZ studia wyższe ukończył na Studium Matematyczno-Technicznym Politechniki Warszawskiej w 1971 r. (w 1976 r. — praca doktorska). Od 1974 r. pracuje w Instytucie Badań Jądrowych, zajmując się zagadnieniami techniki mikrokomputerowej, systemów sterowania i systemu CAMAC.

Mgr inż. Krzysztof RZYMKOWSKI studia ukończył na wydziale elektroniki na Politechnice Warszawskiej w 1968 r. Pracę zawodową rozpoczął w 1967 r. w Zakładzie Doświadczalnym w Biurze Urządzeń Techniki Jądrowej w Warszawie (obecnie Zakład Aparatury Elektronicznej Zjednoczonych Zakładów Urządzeń Jądrowych POLON). Obecnie pracuje w Instytucie Badań Jądrowych (od 1973 r. na stanowisku kierownika Pracowni ds. systemu CAMAC, a od 1977 r. na stanowisku kierownika Pracowni Mikrokomputerów w Zakładzie Elektroniki Jądrowej).

Coraz częściej wymagane są bardzo szybkie urządzenia przetwarzające i dlatego obecnie prowadzone są prace nad szybkim systemem CAMAC. Zastosowanie układów wielkiej skali integracji pozwoli znacznie rozszerzyć możliwości systemu przez budowę zestawów w ramach jednej kasety zawierającej kilka źródeł sterowania.

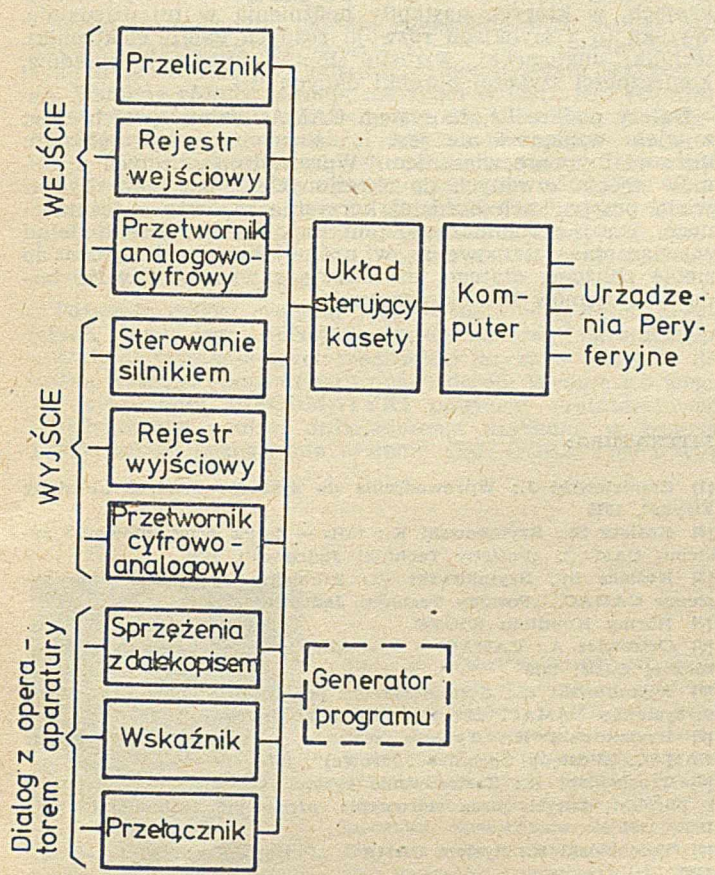
W większości zastosowań wymagających automatycznego sterowania oraz przetwarzania danych można wyróżnić trzy zasadnicze zadania:

1) pomiary, w czasie których następuje gromadzenie i zbieranie danych, a obecnie coraz częściej również ich wstępne przetwarzanie
2) sterowanie, polegające na automatycznej obsłudze dowolnych urządzeń lub wynikające z dialogu pomiędzy urządzeniem sterowanym i sterującym
3) przedstawienie wyników pomiaru w formie dogodnej dla eksperymentatora i umożliwienie przeprowadzenia kontroli procesu sterowania.

System CAMAC może realizować każde z powyższych zadań.

Organizacja systemu jednokasetowego

W kasecie systemu CAMAC znajduje się 25 stanowisk, w których można umieścić bloki o wymiarach około 221 × × 305 × 172 mm, zawierające złącze krawędziowe typu 2 × 43 kontakty, umożliwiające połączenie bloku z magistralą kasety. Magistrala kasety stanowi stałe okablowanie kasety i jest przeznaczona do przesyłania sygnałów pomiędzy blokami wykonawczymi, umieszczonymi w stanowiskach normalnych, a blokiem sterującym, umieszczonym na prawym skrajnym stanowisku kasety nr 25, nazywanym stanowiskiem sterującym. Blok sterujący zajmuje zwykle dwa stanowiska: jedno stanowisko normalne i jedno sterujące (nr 24 i 25). Pozostałe stanowiska są przeznaczone na bloki wejściowe, (bloki pośredniczące, przetworniki analogowo-cyfrowe, bloki sprzęgające), bloki wyjściowe (przetwornik cyfrowo-analogowy, bloki sterowania urządzeń peryferyjnych, bloki sterujące urządzeniami zewnętrznymi, obsługującymi eksperyment), bloki przetwarzające (przeliczniki, konwertery kodów, rejestry, pamięci) oraz ewentualnie na inne bloki specjalistyczne spełniające indywidualne wymagania użytkownika. W systemie używane jest słowo o długości 24 bity. Czas jednorazowego przesłania informacji wynosi 1 μs. Sygnały są przekazywane w logice ujemnej TTL.



Rys. 1. Układ jednokasetowego rozwiązania systemu CAMAC

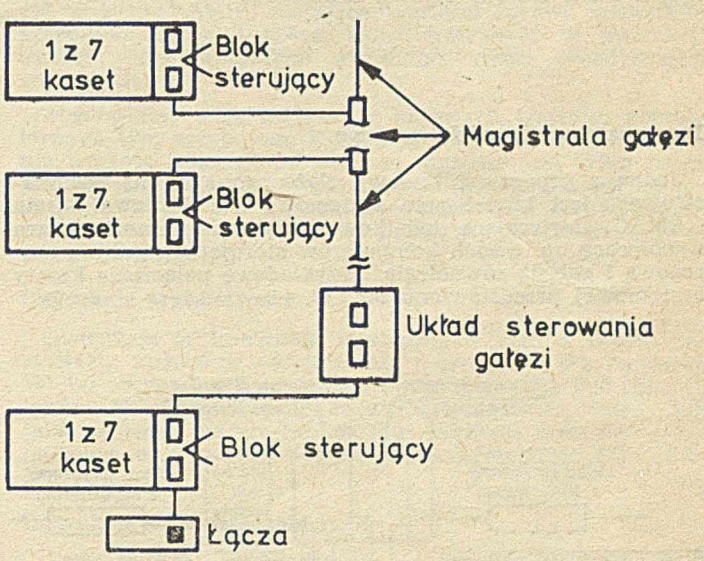
Blokiem sterującym w zestawie jednokasetowym może być blok autonomicznego sterowania kasety (dla zestawów autonomicznych), spełniający rolę komputera sterującego, lub też blok sprzęgający zestaw jednokasetowy z zewnętrznym urządzeniem sterującym. Jednokasetowe systemy CAMAC działają na zasadzie sekwencyjnej realizacji oddzielnych operacji typu „rozkaz" (do bloku wykonawczego) oraz „odpowiedź" (z bloku wykonawczego do bloku sterującego).

Pojedyncza operacja polega na ustawieniu odpowiedniego rozkazu składającego się ze zbioru sygnałów, umożliwiających wybór stanowiska *N*, adres wewnętrzny bloku *A* i kod operacji, którą ma wykonać fragment wywołanego bloku *F*, z danych 24 sygnałów *R* lub *W*, przesyłanych po odpowiednio przyporządkowanych szynach magistrali. Wykonanie rozkazu następuje w chwili przyjścia sygnałów strobujących S_1, S_2.

Spośród 32 możliwych rozkazów przesyłanych po magistrali kasety 16 zostało określonych w zaleceniach normalizacyjnych. Przykład zestawu jednokasetowego przedstawiono na rys. 1.

Organizacja wielokasetowego systemu równoległego

Wielokasetowy system równoległy, nazywany również gałęzią równoległą, polega na połączeniu zestawów jednokasetowych, z których każdy sterowany jest przez sterownik kasety dołączony do magistrali gałęzi równoległej, stanowiącej 132-przewodowe łącze (66 przewodów sygnałów, 66 przewodów powrotu prądu). Magistrala ta przeznaczona jest do przesyłania rozkazów ze sterownika gałęzi równoległej (którym może być komputer) do poszczególnych kaset oraz sygnałów odpowiedzi z kaset do sterownika.



Rys. 2. Układ blokowy systemu równoległego

W systemie równoległym przewidziano możliwości połączenia 7 kaset. Długość przewodów magistrali gałęzi nie może przekraczać 40 m. Ze względu na równoległe przesyłanie sygnałów zawierających wszystkie informacje (adres kasety, kody rozkazów itp. informacje konieczne do wykonania żądanej czynności) system równoległy jest stosunkowo szybki.

Zalecaną konfigurację systemu równoległego przedstawiono na rys. 2.

Organizacja wielokasetowego systemu szeregowego

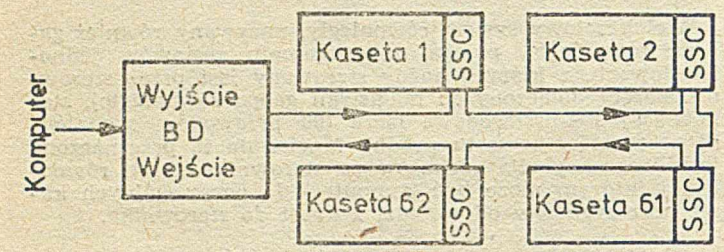
Innym możliwym połączeniem zestawów jednokasetowych jest tzw. gałąź szeregowa, pozwalająca na utworzenie systemu składającego się z 62 kaset, połączonych w zależności od rodzaju transmisji dwoma (przy przesyłaniu informacji w systemie szeregowo-bitowym) lub dziewięcioma (przy przesyłaniu szeregowo-bajtowym) przewodami. Do przesyłania informacji może być wykorzystane normalne łącze telekomunikacyjne, dzięki czemu nie ma praktycznie ograniczeń odległości, w jakiej mogą być rozstawione kasety.

System szeregowy jest nieco wolniejszy od systemu równoległego, a mianowicie czas trwania cyklu jednej operacji w systemie równoległym wynosi ≈1,5 μs, w systemie szeregowym bajtowym — 5 μs, a szeregowym bitowym — 50 μs. W systemie szeregowym poszczególne zestawy jednokasetowe są połączone z magistralą gałęzi szeregowej poprzez szeregowe bloki sterowania kaset, spełniające odpowiednie wymagania transmisji informacji.

System może być sterowany za pomocą komputera lub specjalnego sterownika gałęzi szeregowej. W systemie szeregowym sterowniki kaset połączone są szeregowo i przekazywanie informacji odbywa się jednokierunkowo od 1 do 62 kasety. Układ stanowi zamkniętą pętlę.

W systemie przewidziano metodę wykrywania błędów transmisji. Z ciągu informacji przesyłanych przez magistralę każdy ze sterowników przed wysłaniem informacji do dalszej części magistrali wychwytuje skierowane do niego informacje i w odpowiedniej chwili przekazuje odpowiedź do bloku sterowania gałęzi szeregowej.

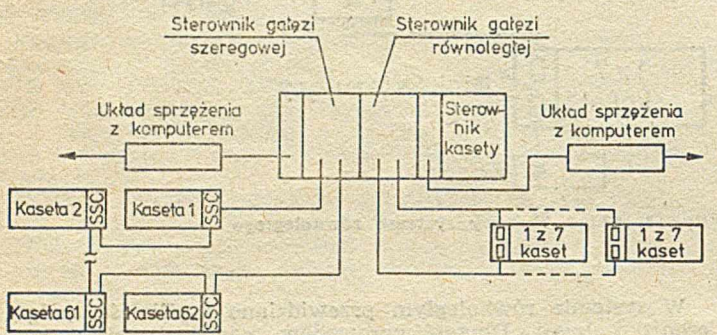
Podstawową konfigurację systemu przedstawiono na rys. 3.



Rys. 3. Układ blokowy systemu szeregowego

Inne rozwiązania wielokasetowe

Jedną z propozycji budowy złożonego systemu wielokasetowego jest tzw. kaseta systemowa (standardowa kaseta CAMAC, zawierająca dodatkowe połączenia), umożliwiająca współpracę np. dwóch komputerów sterujących gałęzią szeregową i gałęzią równoległą. Przykładowe połączenie kasety systemowej przedstawiono na rys. 4.



Rys. 4. Układ kasety wielosystemowej

Całkowicie odmienna propozycja budowy systemu wielokasetowego, nazywana potocznie zestawami o zdecentralizowanej inteligencji (ang. *distributed intelligence*), polega, najogólniej mówiąc, na wprowadzeniu wewnątrz standardowej kasety CAMAC magistrali pomocniczej tzw. ACB *(Auxiliary Control Bus)*, umożliwiającej łączenie kilku źródeł sterowania.

Oprogramowanie zestawów CAMAC jest zwykle wykonywane przez producenta lub poszczególnych użytkowników i dotąd nie ma w tym zakresie żadnych ograniczeń. Obecnie prowadzone są próby dostosowania powszechnie stosowanych języków programowania, np. BASIC, do wymagań systemu, jednakże przeważają rozwiązania oprogramowania indywidualnego.

AKTUALNY STAN PRAC W KRAJU

Od chwili powstania systemu CAMAC Instytut Badań Jądrowych jako członek ESONE przejął rolę koordynatora prac prowadzonych w tym zakresie w różnych ośrodkach krajowych. Przy IBJ działa też zespół branżowej komisji doradczej, opiniujący poszczególne stadia opracowań konstrukcyjnych i programowych, a Branżowy Ośrodek Normalizacyjny przy IBJ prowadzi wszystkie prace normalizacyjne (krajowe i międzynarodowe) w zakresie aparatury CAMAC.

Podstawowym producentem aparatury CAMAC w kraju są Zjednoczone Zakłady Urządzeń Jądrowych POLON — szczególnie Zakłady Urządzeń Przemysłowych (w Krakowie) i Zakład Aparatury Elektronicznej (w Warszawie). Aparatura dostarczana przez tego wytwórcę również za granicę zdobyła w ubiegłym roku tytuł „Juniora Eksportu".

Warto dodać, że w 1978 r. przy Zarządzie Głównym Stowarzyszenia Elektryków Polskich NOT powstał Komitet CAMAC, popularyzujący zastosowanie tej aparatury w różnych dziedzinach gospodarki narodowej.

W ostatnich latach opracowano i uruchomiono produkcję praktycznie wszystkich typów bloków dostępnych w innych krajach, szczególnie uwzględniono potrzeby krajowe, obejmujące bardzo szeroką skalę urządzeń — począwszy od bogatego asortymentu stale modernizowanej mechaniki i zasilaczy do złożonych układów sterujących zestawy jedno- i wielokasetowe.

Równolegle prowadzone są intensywne prace nad oprogramowaniem. Opracowano bogaty asortyment różnych bloków sprzęgających urządzenia peryferyjne i pamięciowe, bloków wykonawczych oraz bloków analogowych. Stworzono już też możliwość sprzężenia praktycznie wszystkich produkowanych krajowych komputerów z systemem. Do najnowszych opracowań bloków autonomicznych sterowania kasety oprócz znanego procesora 130 należy zaliczyć powstałe ostatnio: blok procesora 131 oraz bloki mikrokomputerowe 180 (MKC 80), CMC-80, ABS-80, zawierające mikroprocesory. Wszystkie bloki autonomicznego sterowania mają rozbudowane możliwości programowe, a zwłaszcza zaawansowane oprogramowanie podstawowe.

Przewidywania dotyczące dalszego rozwoju systemu jednoznacznie wskazują, że maksymalne tempo przyrostu produkcji i opracowań nastąpi w latach osiemdziesiątych — i to zarówno w krajach o rozwiniętej elektronice, jak i w krajach, w których nastąpiły opóźnienia w tej dziedzinie. Wynika to z szybkiego rozwoju różnych gałęzi elektroniki, techniki analogowej, wzrostu stopnia integracji układów, konieczności szybszej obróbki danych itp.

Należy podkreślić, że system CAMAC mimo swoich zalet z wielu względów nie jest i nie powinien być jedynym uniwersalnym rozwiązaniem. Wprowadzenie innych systemów specjalizowanych do określonych zadań może być w wielu przypadkach bardziej korzystne zarówno z technicznego, jak i ekonomicznego punktu widzenia. Znane dotąd rozwiązania systemowe są w ogólnej koncepcji bardzo do siebie zbliżone, dlatego też istnieje zawsze możliwość budowy systemów kombinowanych.

LITERATURA:

[1] Chmielewski J.: Wprowadzenie do systemu CAMAC „Postępy Fizyki", 1975
[2] Koślacz St., Rzymkowski K.: IML — język programowania systemu CAMAC. „Postępy Techniki Jądrowej", 1974
[3] Koślacz St., Rzymkowski K.: Problemy oprogramowania systemu CAMAC. „Postępy Techniki Jądrowej", 1974
[4] Normy Komitetu ESONE
[5] Ostrowicz A.: CAMAC — modułowy system aparatury elektronicznej OJEJ, 1976
[6] Rzymkowski K.: Systematyzacja pojęć i terminów używanych w systemie CAMAC. „Postępy Techniki Jądrowej", 1974
[7] Rzymkowski K.: Typowe zestawy pomiarowe w systemie CAMAC. „Postępy Techniki Jądrowej", 1974
[8] Trechciński R.: Zastosowanie systemu CAMAC do rejestracji i obróbki danych oraz sterowania procesami technologicznymi. PIE, 1974
[9] Trechciński R.: System CAMAC. „Postępy Techniki Jądrowej", 1975
[10] Trechciński R.: CAMAC — informacja o wdrożeniach. „Postępy Techniki Jądrowej", 1976

ANTONI MACHOWSKI, WIESŁAW STAROWICZ
Instytut Organizacji i Techniki Transportu Kolejowego
Politechnika Krakowska

Doświadczenia z programowania w języku CSL ODRA 1300

Systemem językowym, w rozumieniu niniejszego artykułu, jest język programowania wraz z oprogramowaniem umożliwiającym jego wykorzystanie na konkretnej maszynie cyfrowej. I tak dla języka symulacyjnego CSL wersji ICL 1900, realizowanego na komputerze ODRA 1300, system językowy CSL ODRA 1300 stanowią, oprócz wspomnianej wersji języka, translatory CSL — programy #XDC3, #XDC4, #XDC5 i #XDC6, system kompilacyjny języka FORTRAN, na którym opiera się CSL, łącznie z grupami bibliotecznymi podprogramów bazowych CSL o nazwach SRC2 i SRC5 oraz pomocnicze programy organizacyjne, np. edytor #XKYA itp.

Proces tworzenia programu wynikowego w tym systemie językowym odbywa się w dwóch zasadniczych fazach: translacji i kompilacji. Translację, polegającą na przetłumaczeniu CSL na program w języku FORTRAN, przeprowadza się jednym z wymienionych programów translujących. Do kompilacji (i konsolidacji) można używać różnych kompilatorów FORTRAN w zależności od użytego translatora CSL (rodzaju nośnika wyjściowego): #XFAM, #XFAE, #XFAT, #XFEW.

Z założenia więc realizację funkcji semantycznych języka CSL zapewnia część logiki translatora CSL, logika podprogramów bazowych CSL oraz część logiki kompilatora FORTRANU wraz z logiką podprogramów systemowych. W organizację tak złożonego procesu tworzenia programu wynikowego włączany jest również z założenia użytkownik opisanego systemu językowego. Ma on do dyspozycji pakiet programów organizacyjnych i instrukcji sterujących pracą translatora CSL, lecz sam musi wybrać i zaprogramować jeden z wielu dostępnych sposobów realizacji tego procesu.

W takim przypadku istnieje szczególnie duża różnica pomiędzy praktyczną znajomością syntaktyki i semantyki języka a praktyczną znajomością całego systemu językowego. Podobne różnice istnieją nie tylko w CSL, lecz również — w mniejszym lub większym stopniu — w innych systemach językowych i wynikają zwykle z niemożności przekazania wszystkich niezbędnych do zrozumienia informacji w stosunkowo krótkich opisach systemów, z błędów drukarskich i tłumaczeniowych powodujących często błędną interpretację znaczeniową, z ewentualnych błędów w oprogramowaniu, jego ulepszeń i zmian dokonywanych przez producentów, itp.

Poznanie całego systemu językowego i wszystkich przyjętych przy jego oprogramowaniu założeń ograniczających bądź rozszerzających możliwości języka w stosunku do opisu, wymaga sporo czasu i doświadczeń. Wydaje się więc, że czytelników INFORMATYKI powinny zainteresować przedstawione poniżej doświadczenia uzyskane w trakcie praktycznego poznawania systemu CSL ODRA 1300 i wykorzystywania go do tworzenia modeli symulacyjnych systemów transportowych. Należy jednak dodać, że poznanie i „przebadanie" systemu językowego określonej generacji nie zwalnia od ciągłego śledzenia zmian wprowadzanych przez producenta w nowych generacjach.

ZASADY OGÓLNE

Istnieje możliwość wprowadzania w segmentach CSL pojedynczych instrukcji i grup instrukcji w FORTRANIE przepisywanych następnie do zbioru wyjściowego przez translator (gdy pierwszym znakiem w rekordzie jest F) bez tłumaczenia.

Wprowadzenie w ten sposób całych segmentów, jak również wierszy sterujących kompilacją FORTRANU (użycie litery F zamiast znaku x), nie jest dopuszczalne i może zakończyć się nawet zapętleniem translatora. Pojedyncze segmenty i grupy segmentów w FORTRANIE oraz półskompilowanych mogą być włączane tylko za pomocą instrukcji SOURCE i SWITCH.

System CSL generuje i umieszcza w programie sekwencję w FORTRANIE, nadającą wartości zerowe wszystkim artybutom stanu i czasu, zadeklarowanym w instrukcji CLASS, oraz wszystkim atrybutom czasu pojedynczych zmiennych.

Zbędne jest więc nadawanie zerowych wartości początkowych tym atrybutom. Konieczne jest natomiast nadawanie wartości różnych od zera, co możliwe jest tylko przez instrukcję podstawienia, gdyż użycie w tym przypadku instrukcji DATA nie ma sensu — sekwencja zerująca CSL zmieni wartości ustawione na podstawie DATA przez program ładujący.

Instrukcje arytmetyczne

Instrukcja arytmetyczna występuje w CSL w dwóch postaciach: instrukcji podstawiania i instrukcji przyrostowej. W obu przypadkach zmienna i wyrażenie mogą być różnych typów. Należy podkreślić, że wystąpienie w wyrażeniu lub jako zmiennej nazwy tablicy (bez indeksu) zostanie zasygnalizowane jako błąd dopiero przez kompilator FORTRANU.

Instrukcje organizowania pętli

Pętlę w CSL organizuje się za pomocą instrukcji FOR i przez wcięcie instrukcji, które mają być tą pętlą objęte. Ostatnią instrukcją w pętli może być instrukcja arytmetyczna IF. Instrukcja organizująca pętlę FOR może występować w dwóch zasadniczych postaciach: z indeksowaniem przyrostowym oraz z indeksowaniem według zbioru.

Mgr inż. Antoni MACHOWSKI ukończył Instytut Transportu Politechniki Warszawskiej w 1972 roku (specjalność: sterowanie ruchem kolejowym). Od 1977 roku pracuje w Instytucie Organizacji i Techniki Transportu Kolejowego Politechniki Krakowskiej. Specjalizuje się w modelowaniu symulacyjnym i analitycznym systemów transportowych. Poza tym interesuje się oprogramowaniem maszyn cyfrowych ODRA 1300.

Dr inż. Wiesław STAROWICZ ukończył studia na Wydziale Elektrotechniki w Akademii Górniczo-Hutniczej w 1971 r. (doktorat — 1977 r.). W latach 1971—1975 pracował w Dyrekcji Okręgowej Kolei Państwowych w Krakowie, od 1976 r. pracuje w Instytucie Organizacji i Techniki Transportu Kolejowego w Politechnice Krakowskiej — ostatnio na stanowisku zastępcy dyrektora ds. badań naukowych. Zajmuje się zastosowaniami informatyki w transporcie kolejowym.

Parametry dla instrukcji FOR pierwszej postaci są takie same jak dla realizującej ją instrukcji DO w FORTRANIE, gdyż CSL nie wprowadza żadnych dodatkowych ograniczeń. Początkowa wartość, końcowa wartość oraz przyrost wartości (krok) wskaźnika pętli mogą być dowolnymi wyrażeniami arytmetycznymi typu całkowitego (a nie jak podano w [1], [2] i [4]), przy czym wartość początkowa może być co najwyżej równa końcowej wartości wskaźnika.

Złożone instrukcje warunkowe

W języku CSL istnieją instrukcje złożone, umożliwiające badanie pewnych relacji zachodzących w zbiorach. W opisach składni tych instrukcji nie zawsze jest podane, że łańcuch badań może być pominięty.

Tak więc mogą być opuszczone łańcuchy badań dla instrukcji ALL, EXISTS, UNIQUE, COUNT, SUM oraz CHAIN; uzyskuje się wtedy dla nich dodatkową funkcję — z wyjątkiem instrukcji ALL oraz CHAIN, które będą wprawdzie poprawnie translowane i wykonywane, lecz nie wnoszą żadnych funkcji użytkowych do programu. Dla instrukcji EXISTS, UNIQUE, COUNT oraz SUM dodatkową funkcją jest działanie dla wszystkich obiektów, a nie tylko dla wyróżnionych zespołem cech opisanym przez łańcuch badań.

Instrukcje generowania liczb pseudolosowych

System CSL daje możliwość generowania liczb pseudolosowych rozkładu równomiernego, normalnego, wykładniczego, Poissona oraz próbkowania z rozkładu dowolnego, podanego w postaci częstościowej, który jest najczęściej rozkładem nie rozpoznanym. Generatory liczb pseudolosowych wymienionych rozkładów są realizowane przez segmenty typu FUNCTION z grupy bazowej CSL (odpowiednio IRAN1, IDEV1, INEG1, IPOS1 oraz ISAM1), wywoływane w programie FORTRANU w odpowiedzi na odpowiednie instrukcje podstawiania CSL.

Często wygodniej jest korzystać z wymienionych funkcji bezpośrednio wywołując odpowiednie segmenty poprzez nazwę w instrukcjach FORTRANU i przez nazwę poprzedzoną znakiem £ w instrukcjach CSL.

W stosunku do standardowego wywołania w CSL liczba i kolejność parametrów nie ulegają zmianie. Początkującym wypada tu przypomnieć, że instrukcja RANDOM (segment IRAN1) generuje liczby pseudolosowe w przedziale od 1 do określonego zakresu włącznie (niezgodność z opisem w [2]) oraz że średnie winny być określone przez wyrażenia typu całkowitego, z wyjątkiem instrukcji POISSON (funkcja IPOS1), dla której średnia powinna być określona wyrażeniem dającym wynik całkowity, lecz w zapisie zmiennoprzecinkowym (typ REAL).

Histogramy

CSL umożliwia zbieranie statystyk z procesu symulacyjnego w postaci wykresów słupkowych (histogramów). Histogramy deklaruje się instrukcją HIST, dystrybucję zdarzeń przeprowadza się instrukcją ADD, zerowanie statystyk instrukcją CLEAR, testowanie (badanie liczby obserwacji w przedziałach) instrukcją YIELD, a wydruk statystyk za pośrednictwem instrukcji OUTPUT. W odpowiedzi na instrukcję HIST generowana jest instrukcja w FORTRANIE rezerwująca odpowiednie tablice oraz sekwencja instrukcji ustawiająca parametry i zerująca histogram. Tak więc bezpośrednio po instrukcji HIST nie ma potrzeby zerowania histogramów instrukcją CLEAR.

Na podstawie szczegółowej analizy wymienionych instrukcji i podprogramów OUTP1 i ADDA1, realizujących instrukcje OUTPUT i ADD, stwierdzono ograniczenie w translatorze, uniemożliwiające podział zakresu liczb ujemnych i wynikające częściowo z założenia [1] a częściowo z błędu w podprogramie ADDA1, nieprawidłowo kwalifikującym wartości ujemne do przedziałów liczbowych. Korzystając więc z tej wersji CSL, należy unikać wartości ujemnych, co nie zawsze jest wygodne i możliwe.

Podprogram OUTP1 — przez analogię do systemu CSL — ICL System 4 — uzupełniono o wydruk histogramu w postaci graficznej z automatycznym wyznaczaniem skali, opcjonalny wydruk nagłówka z nazwą histogramu oraz dodatkowo o samoczynne ograniczanie liczby przedziałów w przypadku zerowych częstości w przedziałach.

Instrukcje diagnostyczne

Diagnostyka własna CSL obejmuje wyprowadzanie:
1) zawartości zmiennych, elementów tablic, atrybutów obiektów, zbiorów
2) danych o realizacji programu w czasie
3) komunikatów o błędach, dotyczących:
   a) przekroczenia przedziału dopuszczalnego zmiennej sterującej w złożonych instrukcjach ADD oraz SAMPLE
   b) przepełnienia zbioru boolowskiego

i podających oprócz rodzaju błędu nazwę segmentu, w którym ten błąd wystąpił.

Diagnostyka CSL opiera się na wielowejściowym segmencie o nazwie MVT1, dołączanym zawsze do programu niezależnie od stopnia jego wykorzystania.

Wyprowadzanie 1) realizowane jest za pomocą instrukcji CHECK oraz PRISET. Instrukcja CHECK jest wyraźnie ograniczona w działaniu, czego nie podkreślono w [1], [2], [3] i [4]. Za jej pomocą można testować jedynie wartości typu całkowitego, co oznacza, że przy użyciu tej instrukcji do wyprowadzania wartości odpowiadających nazwom zadeklarowanym w instrukcji FLOAT wyniki będą niewłaściwe (interpretacja liczb zmiennoprzecinkowych jako całkowitych).

Wyprowadzania 2) oraz 3) są sterowane przełącznikami słowa trzydziestego, przy czym obszar programu, który ma być kontrolowany, wyznacza się instrukcjami START oraz FINISH. Instrukcje te nie mogą występować we wciętych sekcjach programu. Stanowią one dyrektywy dla translatora CSL, wyznaczające odpowiednio początek i koniec włączania sekwencji wywoławczych odpowiednich wejść do segmentu MVT1.

CZĘŚĆ SYMULACYJNA PROGRAMU

W strukturze programu symulacyjnego można wyróżnić trzy zasadnicze części: inicjującą, symulacyjną i finalizującą. W języku CSL początek części symulacyjnej wyznacza się instrukcją ACTIVITIES, która jest dyrektywą nakazującą translatorowi włączenie procedur upływu czasu. Koniec części symulacyjnej wyznaczony jest instrukcją badającą, czy nastąpił koniec symulacji. Fakt ten można badać przez zliczanie liczby cykli symulacyjnych lub kontrolowanie stanu zegara modelowego.

Część symulacyjną można dzielić w CSL na bloki funkcjonalne (bloki działań) za pomocą instrukcji BEGIN. Instrukcję tę można również wykorzystywać w programach niesymulacyjnych, gdyż jej głównym zadaniem jest generowanie adresu dla poprzedniego bloku działań.

W przypadku zakończenia symulacji następuje wykonanie części finalizującej, w której ostatnią instrukcją musi być instrukcja zakończenia wykonywania programu (podprogramu) EXIT RETURN; instrukcja EXIT jest realizowana przez wejście EXIT1 w segmencie MVT1.

W przeciwnym przypadku instrukcja badająca powinna przekazywać sterowanie do instrukcji END lub do instrukcji pustej poprzedzającej END. W segmentach MASTER programów niesymulacyjnych użycie instrukcji EXIT przed END jest zbędne, gdyż instrukcja END w tym przypadku sama generuje wywołanie EXIT1.

Należy podkreślić, że instrukcja END generuje w razie potrzeby adres, podobnie jak instrukcje BEGIN oraz ACTIVITIES (ACTIVITIES może zastąpić pierwszą instrukcję BEGIN). Wygodnie jest więc traktować instrukcję END jako ostatni blok działania, realizujący aktualizację wszystkich atrybutów czasowych i zegara oraz skok do początku części symulacyjnej (END generuje wywołanie wyjścia TIME1 do segmentu MVT1).

Tak więc przykład instrukcji kończącej część symulacyjną można zapisać prościej niż to podano w [1], [2] czy [4]:

```
. . . . . .
. . . . . .

CLOCK GE 1000
. . . . . .

EXIT
END
```

zamiast:

```
. . . . . .  } część symulacyjna
. . . . . .

CLOCK GE 100 a 100 instrukcja badająca
. . . . . .  część finalizująca
EXIT
100 DUMMY
END
```

Jeżeli tworzy się strukturę blokową za pomocą instrukcji BEGIN w programach (segmentach) niesymulacyjnych, to instrukcja END będzie prawidłowo generować adres tylko w segmencie MASTER. W segmentach typu

SUBROUTINE lub FUNCTION w przypadku skoku do END (pustej części adresowej w instrukcji ostatniego bloku działań) nie nastąpi wyjście z tego segmentu, lecz zakończenie pracy programu.

CSL A SYSTEM ŚLEDZENIA TRACE

W opisach CSL ODRA 1300 niezbyt jasno przedstawiono możliwości wykorzystania systemu śledzenia FORTRANU. I tak, w [1] z jednej strony nakazuje się bezwzględnie użycie wiersza TRACE 0, natomiast w przykładzie programu używa się poziomu TRACE 2. Po analizie problemu stwierdzono, że każdy z poziomów śledzenia TRACE może być wykorzystywany przy uwzględnieniu niżej opisanych ograniczeń.

Poziom TRACE 0 musi być użyty, jeśli w którymkolwiek segmencie CSL wykorzystuje się histogramy. Wynika to z tego, że CSL generuje w tym przypadku sekwencje w FORTRANIE używające indeksu 0 do wyznaczania adresu zerowego tablic.

Poziom TRACE 0 należy również bezwzględnie zastosować w programie składającym się z więcej niż jednego segmentu napisanego w CSL, jeśli w nazwie segmentu CSL wywołującego inny segment CSL piąty znak nie jest cyfrą ani spacją. Ograniczenie to wynika z użycia instrukcji podstawiania w FORTRANIE do przesłania ośmiu pierwszych znaków nazwy segmentu wywołującego.

Nieuwzględnienie tych dwóch zasadniczych warunków spowoduje wystąpienie błędów wykonania, odpowiednio 1 (przekroczony zakres wskaźnika) oraz 50 (nadmiar — zapalone V).

Czas modelowy a struktura wielosegmentowa

Zmienna o nazwie CLOCK z założenia reprezentuje czas na zegarze modelu. W segmencie zawierającym część symulacyjną organizującą proces symulacyjny wartość zegara dostępna jest poprzez nazwę CLOCK w instrukcji CSL i przez nazwę 1000 w instrukcji FORTRANU.

System CSL nie zapewnia jednak przekazania wartości zegara innym segmentom. Użytkownik może więc w razie potrzeby zadbać o przekazanie tej wartości parametrycznie lub przez obszar wspólny (COMMON).

Wykorzystanie urządzeń wyjściowych

Wszystkie segmenty CSL używają do wyprowadzania informacji drukarki wierszowej o numerze programowym 0, przydzielanej statycznie, tj. po przeczytaniu bloku żądań (dyrektywa PLANU PERIPHERAL). Jeśli konieczne jest umieszczenie użytkowych wyników na oddzielnym tabulogramie, a do dyspozycji programu są dwie drukarki, należy to uwzględnić w wierszach sterujących dla kompilatora FORTRANU poprzez określenie innego numeru programowego.

Instrukcje sterujące translatorem CSL

Instrukcje sterujące operacjami organizacyjnymi translatora CSL umożliwiają wybór nośnika wejściowego, określają typ segmentu lub segmentów (CSL, FORTRAN, SEMI-COMPILED) wejściowych, dokonują wyboru struktury zbioru wyjściowego itp.

W przypadku użycia instrukcji SOURCE do określenia wejścia z taśmy magnetycznej należy mieć na uwadze dość istotne ograniczenia. Mianowicie, przy wcześniejszym zapisywaniu programu CSL na taśmę magnetyczną (np. za pomocą edytora #XKYA), oprócz zapewnienia struktury podzbioru o określonej nazwie z wpisanym w dwunaste słowo oznacznika początku podzbioru wyróżnikiem typu segmentu (dla CSL-B2S1), należy bezwzględnie wymusić zapis zawartości jednej karty (80 znaków taśmy dziurkowanej) w jednym bloku. Osiąga się to przez ustalenie zblokowania (instrukcją BLO dla #XKYA) na 21 słów oraz przez uzupełnienie rekordów na taśmie dziurkowanej spacjami do wspomnianych 80 znaków, bądź też dziurkowanie w co najmniej jednej z ostatnich czterech kolumn karty znaku różnego od spacji. Niespełnienie powyższych warunków spowoduje całkowicie błędną interpretację programu źródłowego przez translator CSL.

Parę z instrukcją SOURCE stanowi instrukcja SWITCH, powodująca przełączenie czytania na określone przez SOURCE urządzenie. W przypadku zbioru na taśmie magnetycznej dojście w trakcie czytania do końca podzbioru powoduje — w zależności od wejścia do translatora — samoczynne przełączenie na czytnik kart (CR) lub czytnik taśmy dziurkowanej (TR).

Para instrukcji OBJECT i ENDSUBFILE, oprócz funkcji ustalenia rodzaju nośnika i nazwy zbioru realizowanej przez OBJECT, służy odpowiednio do zapisu oznacznika początku i końca podzbioru. Brak instrukcji ENDSUBFILE nie zawsze jest kontrolowany. Należy szczególnie uważać, aby nie brakowało tej instrukcji przed końcem zbioru CSL.

Opisany system edycji umożliwia tworzenie bibliotek segmentów półskompilowanych oraz źródłowych zarówno w CSL, jak i w FORTRANIE.

OGRANICZENIA SYSTEMU CSL

Większość ograniczeń CSL jest pozorna i możliwa do zlikwidowania przez wprowadzenie dodatkowych instrukcji zapisanych w FORTRANIE (np. typ LOGICAL i DOUBLE PRECISION może być określony przez dodanie po wstępnej translacji dyrektyw typu). Ograniczenie nie dopuszczające stosowania stałych typu TEXT jako parametrów aktualnych można zlikwidować tylko w przypadku, gdy w stałych tych nie występują spacje. Można uzyskać to przez użycie znaku £.

I tak instrukcja

```
CALL £ COPY (5, NAME (1), £ 5 HALFAK, 1)
```

będzie przetłumaczona np. na

```
CALL COPY (5, 1002 (1), 5 HALFAK, 1)
```

z czego widać, że trzeci parametr aktualny przetłumaczony jest przez translator prawidłowo.

KORZYSTANIE Z CSL W SYSTEMIE GEORGE 2

Ze względu na brak standardowych macro, autorzy we własnym zakresie opracowali trzy makrodefinicje realizujące następujące funkcje:

1) CSL
— aktualizacja programów źródłowych na TM za pomocą #XKYA
— translacja za pomocą wybranego translatora
— kompilacja, konsolidacja i uruchomienie programu za pomocą macro FORTRAN

2) CSLMT
— translacja z TM za pomocą wybranego translatora (pozostałe jak w CSL)

3) CSLCR
— translacja za pomocą wybranego translatora (pozostałe jak w CSL).

Sposób ich wywołania jest następujący:

CSL TSNwe, TSNwy, TAPEwe, TAPEwy, lab1, tran, lab2, ident, pd, label, time, comp

CSLMT TSNwe, TAPEwe, lab1, tran, lab2, ident, pd, label, time, comp CSLCR tran, lab2, ident, pd, label, time, comp

gdzie:

TSNwe, TSNwy, TAPEwe, TAPEwy oznaczają odpowiednio numery seryjne oraz etykiety taśm wejściowych i wyjściowych

lab1, lab2 — etykiety wyjścia z macro w przypadku błędów (odpowiednio) aktualizacji i translacji

tran — wybrany translator CSL (dyskowy lub taśmowy)

ident, pd, label, time, comp — znaczenie takie jak w macro FORTRAN

Makrodefinicje opisane w systemie GEORGE zostały wprowadzone do zbioru standardowych macro w ośrodku obliczeniowym w Politechnice Krakowskiej.

* * *

Przedstawione w artykule informacje wynikają z doświadczeń uzyskanych podczas pracy z translatorem taśmowym #XDC3 MK7 oraz z translatorami dyskowymi #XDC5 MK6 i #XDC6 MK1. Nie wyczerpują one oczywiście wszystkich problemów, z jakimi może mieć do czynienia użytkownik po zapoznaniu się jedynie z ogólnodostępnymi opisami zasad języka. Mamy jednak nadzieję, że przekazane uwagi pomogą zainteresowanym w opanowaniu programowania zagadnień symulacyjnych.

LITERATURA:

[1] Język do modelowania układów zdarzeń CSL WZE ELWRO. Zeszyt 13102

[2] Kondratowicz L.: Modelowanie symulacyjne systemów. WNT 1978

[3] Kondratowicz L.: Symulacja cyfrowa w języku CSL. PWN 1974

[4] Wajs W.: Język do symulacji i kontroli CSL. Skrypt AGH 1976

BOGDAN STEFANOWICZ
Szkoła Główna Planowania i Statystyki
Warszawa

Metody heurystyczne w przetwarzaniu danych

Nowe zastosowania komputerów, oparte na metodach tzw. sztucznej inteligencji, świadczą o wielu nie wykorzystanych jeszcze możliwościach zautomatyzowania najrozmaitszych operacji. Metody te prezentowane na łamach INFORMATYKI w cyklu „Sztuczna Inteligencja", można podzielić na trzy główne grupy [1]). Do pierwszej będą należały metody pozwalające przedstawić w pamięci komputera zebrane informacje o określonych obiektach, faktach itp. w sposób umożliwiający przeprowadzenie szczegółowych i kompleksowych analiz. Są to tzw. metody prezentacji wiedzy. Grupę drugą stanowią metody pozwalające zaprogramować „uczenie się" komputera, tzn. gromadzenie przezeń w sposób systematyczny i celowy wszystkich takich informacji, które mogą okazać się niezbędne lub przydatne w dalszej pracy maszyny. Wreszcie do ostatniej grupy można odnieść metody nietrywialnego, nieszablonowego rozwiązywania zadań stawianych przed maszyną. Metody takie noszą nazwę heurystycznych.

Wiele jest przykładów świadczących o przydatności metod heurystycznych. Rzadko jednak stosuje się je w procesach projektowania, budowania i eksploatacji „typowych" systemów przetwarzania danych i systemów informatycznych. Tym bardziej celowe wydaje się przytoczenie kilku przykładów efektywnego stosowania tych metod w praktyce.

PRZYKŁAD PIERWSZY

Konieczność sprawdzania poprawności danych podlegających przetwarzaniu jest oczywista i bezsporna. Obecnie znane są i stosowane rozmaite metody i sposoby wykrywania błędów w danych i ich korygowania. W zasadzie jednak nie zawsze jest możliwe lub celowe wykrycie i usunięcie wszystkich błędów, jakie mogą się pojawić w opracowywanym materiale wejściowym [2]). Przykładem są badania i analizy statystyczne, w których poszczególne dane jednostkowe nie mają istotnego wpływu na wyniki końcowe. W takich przypadkach koszty związane z wykrywaniem i usuwaniem najdrobniejszych nawet pomyłek mogą okazać się nieporównywalnie wyższe od osiąganej (często pozornej) dokładności rezultatów przetwarzania. Już sam czas potrzebny na wykrycie błędów, ich wyjaśnienie i skorygowanie powoduje znaczne opóźnienia w dostarczeniu potrzebnych użytkownikom informacji. Warto też odnotować, że błędy w danych statystycznych, jak zresztą i w innych danych, noszą charakter względny. Błędna cyfra w pozycji jednostek w jakiejś liczbie nie będzie miała żadnego znaczenia przy podawaniu odpowiednich wielkości z dokładnością do setek czy tysięcy. Będzie ona natomiast dużo znaczyć, jeżeli liczba ta musi być podana z dokładnością do kilku znaków po przecinku.

Wszystkie te okoliczności powodują, że w procesie sprawdzania poprawności danych podlegających opracowaniu zachodzi konieczność znalezienia takiego rozwiązania, które z jednej strony zapewni wykrycie błędów uznawanych w danym wypadku za poważne, z drugiej, zaś, nie spowoduje zbędnego analizowania wszystkich przypadków „podejrzanych", gdy w istocie nie kryje się w nich żaden duży błąd. Jest to problem, którego rozwiązanie metodami algorytmicznymi jest niemożliwe. Wynika on w szczególności z tego, że szereg podstawowych decyzji podejmuje się w warunkach niepełnej informacji. Po pierwsze, bowiem, trudno jest jednoznacznie i w sposób ścisły (algorytmiczny) ustalić granicę między błędami „dużymi" i „małymi", ponieważ często nie są nawet znane potrzeby przyszłych użytkowników, dla których przygotowywane są określone wyniki; po drugie, brakuje zazwyczaj niezbędnych informacji pozwalających wybrać najskuteczniejszą i najbardziej ekonomiczną metodę kontroli i korekty, tak, aby nie zostały poniesione nadmierne koszty.

Pozostaje więc stosowanie metod heurystycznych, które umożliwiłyby rozwiązanie tych wszystkich kwestii. Przy tym wykorzystywane do tego celu heurystyki można podzielić na dwie grupy. Do pierwszej będą należały te z nich, które umożliwiają znalezienie jakiegokolwiek zadowalającego rozwiązania. Drugie natomiast pozwalają doskonalić tok postępowania w trakcie procesu weryfikacji. Pierwsze można nazwać heurystykami statycznymi, drugie — dynamicznymi. Pierwsze — w warunkach automatyzacji weryfikacji z wykorzystaniem komputerów — stwarzają podstawy do opracowania algorytmu kontroli i korekty oraz odpowiedniego programu maszynowego. Heurystyki dynamiczne natomiast wskazują, jak należy opracować ów program, by możliwe było doskonalenie przebiegu weryfikacji w toku pracy maszyny.

Przykładem heurystyki statycznej jest założenie, że jeżeli jakaś wartość jest mało prawdopodobna, to w przypadku jej odczytania jako danej wejściowej należy upewnić się, czy jest to wielkość poprawna. Cechą charakterystyczną procedur zbudowanych na podstawie tego założenia jest to, że aczkolwiek nie są one w stanie wykryć wszystkich błędów, to mogą jednak wykryć błędy mające istotny wpływ na ostateczne wyniki przetwarzania. Poza tym — druga cecha charakterystyczna tych procedur — przyjęte założenie jedynie formułuje warunek poprawności danych i nie wpływa na przebieg weryfikacji. Raz opracowany na ich podstawie program maszynowy może być niezmiennie realizowany przez komputer w czasie sprawdzania całego materiału — nawet wtedy, gdy towarzyszące temu procesowi okoliczności świadczą o możliwości jego zracjonalizowania.

Możliwość takiej racjonalizacji wskazują heurystyki dynamiczne, np. dynamiczna kontrola i dynamiczna korekta w zakresie weryfikacji danych. Istota tej metody przedstawia się następująco [3]). W celu wykrycia i odrzucenia do korekty mało prawdopodobnych wartości określonej zmiennej przyjmuje się pewne granice, zwane granicami akceptacji. Jeżeli zaobserwowana konkretna wartość tej zmiennej mieści się w owych granicach, traktowana jest jako poprawna; w przeciwnym razie uznawana jest za błędną. Liczba danych zakwestionowanych przez taką procedurę zależy od trafności doboru granic akceptacji. Zwykle ustala się je na podstawie poprzednio przeprowadzonych badań lub innych zebranych informacji. Wiadomo jednak, że raz wyznaczone granice nie mogą być niezmiennie wykorzystywane przez dłuższy czas. Ulegają bowiem zmianie zjawiska, których one dotyczą. Toteż w celu ich zauktualizowania w toku weryfikacji materiału wejściowego ustalone na początku granice są na bieżąco modyfikowane z uwzględnieniem wszystkich danych uznanych przez kontrolę jako poprawne. W ten sposób następuje aktualizacja kryteriów poprawności i dopasowanie ich do faktycznych warunków przetwarzania z uwzględnieniem wszystkich zmian i tendencji, które dają się zaobserwować w badanym zjawisku. Programy wykorzystujące takie procedury „uczą się" — gromadzą informacje o strukturach i własnościach przetwarzanych danych oraz posługują się tymi informacjami w dalszej pracy. Realizują więc swoje zadania efektywniej niż programy pozbawione takiej możliwości: Są to programy heurystyczne, zdolne do automatycznego doskonalenia procesów, które muszą realizować.

---

[1]) Por. Stefanowicz B.: Sztuczna inteligencja. INFORMATYKA nr 7—8/1976

[2]) Por. Buśko B. i Śliwieński J.: Jakość informacji w systemach informatycznych oraz Buśko B.: Ogólna charakterystyka błędów i przyczyn ich powstawania w systemach informatycznych. INFORMATYKA nr 4/1976

[3]) Bliższy opis tych metod zawiera praca B. Stefanowicza: Redagowanie jednostkowych danych statystycznych. GUS—ZMiAOS 1970, zeszyt 3

PRZYKŁAD DRUGI

Dane podlegające przetwarzaniu często są wstępnie porządkowane według określonych kluczy (tzw. sortowanie). Okazuje się, że tę nader czasochłonną czynność można niekiedy wyeliminować przez zastąpienie jej kilkakrotnym czytaniem tego samego zbioru. Na przykład przy opracowywaniu zestawień obrazujących zjawiska demograficzne, w których należy uwzględnić podział materiału wejściowego na dwa podzbiory w zależności od płci badanych osób, wystarczy dwukrotne czytanie danych jednostkowych. Przy pierwszym przebiegu następuje opracowanie tylko tych informacji, które dotyczą kobiet, przy drugim — informacji odnoszących się do mężczyzn. Rozwiązanie takie jest o wiele mniej czasochłonne niż tradycyjne sortowanie całego materiału wejściowego. Jest ono korzystne nie tylko w tym jednym przypadku, lecz zawsze wtedy, gdy bierze się pod uwagę klucz posiadający niewielką liczbę różnych wartości.

Gdy wartości takich jest więcej, można zastosować nieco inne rozwiązanie. Podczas pierwszego czytania opracowywanych danych uwzględnia się tylko te zapisy, które odpowiadają wartościom klucza przypuszczalnie najczęściej powtarzającym się w materiale. Pozostałe są przepisywane do zbioru roboczego (na przykład na pomocniczą taśmę magnetyczną). Przy kolejnym przebiegu wykorzystywany jest ten właśnie zbiór roboczy, z którego do dalszego opracowania pobierane są znów tylko zapisy związane z najczęściej powtarzającymi się wartościami klucza w ramach tego zbioru. Teraz może to być ustalone bardzo dokładnie: niezbędne informacje można uzyskać podczas pierwszego czytania całego zbioru sporządzając odpowiednie statystyki obrazujące rozkład zapisów według wartości rozpatrywanego klucza. Pomijane w tym przebiegu dane znów są przepisywane na inną taśmę roboczą itd. Doświadczenia wskazują, że zastąpienie zwykłego sortowania dużego zbioru danych przez takie właśnie stopniowe jego dzielenie na podzbiory w zależności od wartości klucza w poszczególnych zapisach może przynieść znaczne oszczędności w czasie, jaki jest zwykle potrzebny na wykonanie całej pracy.

Ma to miejsce wówczas, gdy uwzględniany jest jeden klucz sortowania, a rozkład danych według niego nie jest równomierny, tzn. jednym jego wartościom odpowiada więcej zapisów, innym natomiast mniej. Przy czym różnice te są znaczne.

W wielu jednak przypadkach nie da się uniknąć pełnego sortowania. Można wtedy próbować skrócić czas jego wykonania czy to przez połączenie tej operacji z poprzednio naszkicowaną procedurą, czy też przez opracowanie specjalnego programu wykorzystującego częściowe ułożenie materiału wejściowego według wymaganej kolejności. Z obserwacji bowiem wynika, że najczęściej porządkowane dane nie są „losowo" przemieszane i wykazują określone ułożenie według pewnego kryterium. W niektórych przypadkach może to być wykorzystane do przeprowadzenia pożądanego sortowania. Ponieważ jednak z góry nie można tego przewidzieć, przed rozpoczęciem właściwej operacji porządkowania celowe jest sprawdzenie, czy musi ono dotyczyć całego materiału. Niekiedy można tego dokonać na podstawie próbki pobranej ze zbioru wejściowego. Jeżeli wynik takiego badania wykaże „określone ułożenie" danych, trzeba to uwzględnić w procesie sortowania.

Przedstawione rozwiązania mają wspólną cechę: wymagają wykrycia i uwzględnienia określonych charakterystyk sortowanego materiału. W niektórych przypadkach potrzebne na ten temat informacje mogą być zebrane przez samą maszynę zanim przystąpi do wykonywania zasadniczej operacji. Te dodatkowe czynności pochłaniają też dodatkowy czas. Tak więc ostateczny czas realizacji całej operacji nie zawsze jest krótszy od czasu pełnego, klasycznego sortowania. W pewnych przypadkach jest on jednak znacznie krótszy.

W każdej konkretnej sytuacji ponoszone jest pewne ryzyko związane ze stosowaniem przedstawionych metod. Są to bowiem rozwiązania typu heurystycznego, pozbawione cech rozwiązań deterministycznych, algorytmicznych.

INNE PRZYKŁADY

Budowa i eksploatacja systemów informatycznych, zwłaszcza dużych i przeznaczonych do obsługi użytkowników o zróżnicowanych zainteresowaniach, wiążą się z wieloma problemami. Wśród nich można wymienić problemy spójności systemów, porównywalności danych zbieranych w różnych okresach i przy zastosowaniu różnych metod, określenia rodzaju i zakresu informacji, jakie muszą znaleźć się w bazie danych systemu, problemy wyszukiwania danych zgodnie z aktualnymi potrzebami odbiorców itd.

Złożoność tych problemów uniemożliwia ich rozwiązywanie metodami algorytmicznymi. Pozostają więc metody heurystyczne, które pozwalają człowiekowi pokonywać analogiczne trudności napotykane w życiu codziennym. One to prowadzą do znalezienia kompromisu w stosowaniu określonych klasyfikacji i nomenklatur, w określaniu potrzeb użytkowników systemów informatycznych itd.

Metody te pozwalają nie tylko rozwiązywać zasadnicze kwestie projektowania i eksploatacji systemów informatycznych, lecz także doskonalić cząstkowe procesy przetwarzania danych. Tak na przykład F. Wyman opracował program szacowania pewnych parametrów w modelu regresji liniowej, oparty na metodzie heurystycznej, który zużywa zaledwie 10—15% czasu potrzebnego do oszacowania tychże parametrów metodą algorytmiczną. Przy tym wyniki uzyskiwane przez program heurystyczny są obarczone błędem nie przekraczającym 1—4% wartości dostarczanych przez algorytm optymalizacyjny [4]). Innym podobnym rozwiązaniem jest program A. B. Jambekara. Jest to program rozwiązujący problem transportowy, w którym wykorzystano procedurę heurystyczną umożliwiającą uwzględnianie dodatkowych warunków określanych przez użytkownika [5]).

PODSUMOWANIE

Przytoczone przykłady nie są jedynymi, jakie mogą potwierdzić przydatność metod heurystycznych w procecach „zwykłego" przetwarzania danych — czy to w warunkach systemów informatycznych, czy poza nimi. W większości jednak przypadków zastosowania, jakie metody te znajdują w przetwarzaniu, nie wynikają ze świadomego ich wyboru przed przystąpieniem do organizowania, projektowania i realizacji systemów przetwarzania danych; wymuszają je problemy rodzące się w trakcie tych procesów. W takiej sytuacji trudno o pełną ocenę zalet i wad metod, co nie przyczynia się do ich rozpowszechnienia. Konieczne stają się więc systematyczne i zorganizowane badania dotyczące zastosowania metod heurystycznych w przetwarzaniu danych — w pracach projektantów i programistów. W szczególności celowe jest podjęcie badań nad heurystykami szczegółowymi — zarówno statycznymi, jak i dynamicznymi. Prace takie wymagają dokładnego poznania procesów zbierania, przechowywania, opracowywania i dystrybucji informacji dla różnorodnych użytkowników; wymagają też analizy istniejących systemów i wdrożonych rozwiązań systemowych, programowych i organizacyjnych — w celu wyłonienia prawidłowości wynikających ze stosowania określonych metod i sposobów. Informacji takich mogą dostarczyć same programy i systemy, wystarczy tylko wkomponować w nie odpowiednie procedury sporządzające niezbędne statystyki i przygotowujące materiał do wyprowadzenia ostatecznych wniosków. Obiecująco przedstawiają się też metody symulacyjne. Dzięki nim możliwe jest zbadanie różnych wariantów planowanych rozwiązań i wydanie decyzji co do celowości stosowania określonych heurystyk [6]).

[4]) Wyman F. P.: Binary programming: a decision rule for selecting optimal vs heuristic techniques. „The Computer Journal" vol. 16, nr 2/1973

[5]) Jambekar A. B.: A multipurpose heuristic for a location problem. „Bul. Oper. Res. Soc. Am." vol. 23, 1976

[6]) Nader interesujący przykład stosowania metody symulacyjnej do ustalania najwłaściwszego rozwiązania w zakresie weryfikacji danych statystycznych podaje S. Nordbotten w referacie: The efficiency of automatic detection and correction of errors in individual observations as compared with other means for improving the quality of statistics (35 Sesja Międzynarodowego Instytutu Statystycznego, Belgrad 1965)

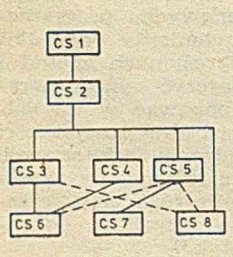
CURRICULUM'78 – program kształcenia informatyków w szkołach wyższych

Pierwszy raport na temat nauczania przedmiotów informatycznych w szkołach wyższych, Curriculum'68, ukazał się w 1968 r. [1]. Gwałtowny rozwój informatyki w ostatnim dziesięcioleciu spowodował jednak wiele istotnych zmian w systemie kształcenia studentów. Zmiany te skłoniły Komitet C3S (*Curriculum Committee on Computer Science* — Komitet ds. Programu Studiów Informatyki) [2] do opracowania nowych, zaktualizowanych zaleceń dotyczących zakresu i metod kształcenia. Powstał kolejny — drugi — raport, wydany w ubiegłym roku, Curriculum'78 [3]. Zgodnie z intencją Komitetu publikacja ta powinna pomóc środowiskom akademickim w dalszym doskonaleniu programów nauczania informatyki w szkołach wyższych, powinna zachęcić je do wymiany opinii i doświadczeń.

Curriculum'78 zawiera przesłanki zalecanego programu (część 1), opis programu bazowego, wspólnego dla wszystkich programów informatyki w szkołach wyższych (część 2), przegląd zagadnień z dziedziny informatyki, zalecanych jako uzupełniające program bazowy (część 3), optymalny program kształcenia w szkołach wyższych, obejmujący przedmioty zarówno z dziedziny informatyki, jak i uzupełniające, łącznie składające się na program studiów na poziomie magisterskim (część 4), przedmioty zalecane w powszechnym systemie kształcenia, systemie wspomagającym, w ramach szkolenia ustawicznego (część 5) oraz wnioski określające warunki (sprzęt, kadra) konieczne do realizacji zalecanego programu kształcenia, a także jako odniesienie do programu studiów dwuletnich (zwłaszcza tzw. programów przejściowych) i programów szkół średnich (część 6).

PROGRAM BAZOWY

Komitet C3S wyodrębnił obszary wspólne dla wszystkich programów informatyki na poziomie podstawowym. Ten blok ośmiu przedmiotów, nazywany programem bazowym, zaleca się włączyć przede wszystkim (choć nie wyłącznie) do programu studiów na kierunku „Informatyka". Realizacja programu bazowego musi poprzedzać wszelkie kursy wyższych poziomów. Przykładowe zestawienie przedmiotów w ramach programu bazowego zawiera rysunek 1.



Rys. 1. Struktura programu bazowego

Program bazowy został tak pomyślany, aby słuchacze po zaliczeniu wybranej grupy przedmiotów posiadali umiejętność sprawnego pisania poprawnych i dobrze udokumentowanych programów oraz określenia poprawności napisanego programu, umieli rozpoznać, które problemy mogą być rozwiązywane środkami informatyki, dobrze znali podstawowe architektury komputerów. Z takim przygotowaniem można podjąć dalsze studia w jednym lub wielu obszarach zastosowań lub specjalizacji na kierunku „Informatyka".

W ramach programu bazowego wydzielono:

- materiał poziomu elementarnego, obejmujący takie zagadnienia, jak:
  — podstawy programowania (algorytmy, języki programowania, techniki programowania, weryfikacja programów, zastosowania)
  — organizacja oprogramowania (struktura komputera i jęmów komputerowych, projektowanie logiczne, reprezenzyki maszynowe, reprezentacja danych, kodowanie symboliczne i systemy scalania, techniki adresowania, makroinstrukcje, segmentowanie programu i łączenie, kompilatory, systemy i programy zastosowaniowe)
  — organizacja sprzętu komputerowego (organizacja systetacje danych: przenoszenie, arytmetyka binarna, przechowywanie binarne i metody dostępu, sterowanie WE/WY, niezawodność)
  — struktury danych i przetwarzanie zbiorów, (sortowanie i przeszukiwanie, drzewa, terminologia zbiorów dostęp sekwencyjny, dostęp bezpośredni, zbiory WE/WY)
  wraz z przykładowym zestawem pięciu przedmiotów [4]
  CS1 [5] — programowanie komputerów I (2-2-3) [6]
  CS2 — programowanie komputerów II (2-2-3)
  CS3 — wprowadzenie do systemów komputerowych (2-2-3)
  CS4 — wstęp do organizacji komputera (3-0-3) lub (2-2-3)
  CS5 — wstęp do przetwarzania zbiorów (3-0-3)

oraz

- materiał poziomu średniego, obejmujący takie zagadnienia, jak:
  — kilka języków programowania
  — architektura maszyn i systemów operacyjnych
  — struktury danych
  wraz z przykładowym zestawem trzech przedmiotów
  CS6 — systemy operacyjne i architektura komputerów (2-2-3)
  CS7 — struktury danych i analiza możliwości algorytmizowania (3-0-3)
  CS8 — organizacja języków programowania (3-0-3).

PRZEDMIOTY DODATKOWE I SPECJALISTYCZNE

Raport zawiera również listę przedmiotów, które można dodatkowo wybrać w czasie studiów zarówno na poziomie podstawowym jak i zaawansowanym. Przedmioty wybrane na poziomie podstawowym studiów nie muszą mieć ścisłego związku z przyjętą specjalizacją. Warunek ten powinien być spełniony w przypadku wyboru dokonanego na bardziej zaawansowanym poziomie studiów.

W odniesieniu do przedmiotów poziomu podstawowego pożądane jest stworzenie możliwości wyboru języków programowania (np. APL, COBOL, LISP, PL/1), najbardziej przydatnych w określonych obszarach zastosowań. Zajęcia powinny odbywać się w laboratorium komputerowym.

Na wyższym poziomie studiów zaleca się do wyboru dziesięć przedmiotów:
CS9 — komputery i społeczeństwo (3-0-3)
CS10 — systemy operacyjne i architektura komputera II (2-2-3)
CS11 — projektowanie systemów zarządzania bazą danych (3-0-3)

---

[1] Curriculum Committee on Computer Science (C3S): Curriculum'68, Recommendations for Academic Programs in Computer Science. Comm. ACM 11,3 (March 1968)

[2] Komitet C3S działa w ramach ACM (*Association for Computing Machinery* — Stowarzyszenie ds. Maszyn Matematycznych)

[3] Curriculum Committee on Computer Science (C3S): Curriculum'78, Recommendations for the Undergraduate Program in Computer Science. Comm. ACM 22,3 (March 1978)

[4] Na zajęcia z każdego przedmiotu przewiduje się trzy semestry

[5] CS to akronim ang. *computer science* (informatyka)

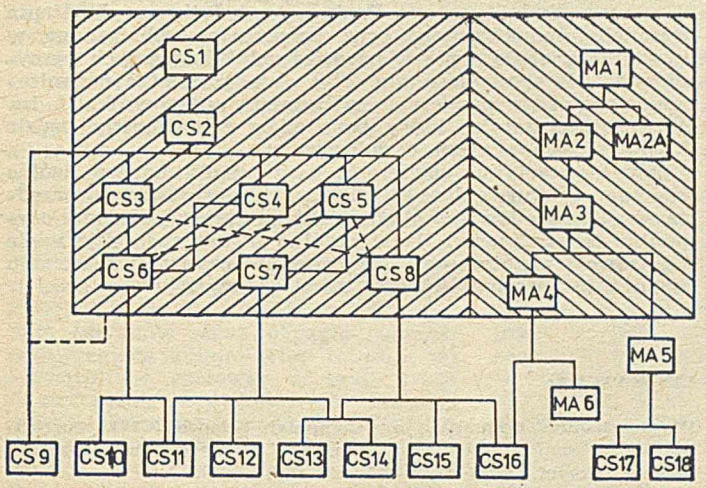
[6] W nawiasie podano liczbę godzin, jakie należy przeznaczyć w ciągu tygodnia kolejno na: wykłady, ćwiczenia, zajęcia w laboratorium komputerowym. I tak np. (2-2-3) oznacza, że 2 godziny przewiduje się na wykłady, 2 godziny — na ćwiczenia i 3 godziny — na zajęcia w laboratorium komputerowym

CS12 — sztuczna inteligencja (3-0-3)
CS13 — algorytmy (3-0-3)
CS14 — projektowanie i rozwój oprogramowania (3-0-3) lub (2-2-3)
CS15 — teoria języków programowania (3-0-3)
CS16 — automaty, obliczalność i języki formalne (3-0-3)
CS17 — metody numeryczne: analiza (3-0-3)
CS18 — metody numeryczne: algebra liniowa (3-0-3).

Przedmioty te wymagają stworzenia odpowiedniej kadry dydaktycznej, niestety niewiele szkół wyższych dysponuje takimi możliwościami.

Zaleca się także wprowadzenie przedmiotów specjalistycznych do programu studiów na kierunku „Informatyka". Materiał przekazywany w ramach tych przedmiotów powinien być wysoce aktualny i ściśle związany ze specjalizacją. W miarę rozwoju dyscypliny przedmioty specjalistyczne mogą wzbogacać programy poziomów niższych.

Oto przykładowo zagadnienia, które należałoby uwzględnić w programie specjalizacji:
— laboratorium mikrokomputerowe
— laboratorium minikomputerowe
— ocena sprawności
— zagadnienia telekomunikacji (sieci komputerowe), systemy rozproszone
— symulowanie systemów
— programowanie systemów rozwiniętych
— grafika komputerowa
— laboratorium: budowanie kompilatorów
— programowanie strukturalne
— wybrane zagadnienia teorii automatów
— wybrane zagadnienia obliczalności *(computability)*
— wybrane zagadnienia teorii języków formalnych
— symulacja i modelowanie.



Rys. 2. Zalecana struktura programu kształcenia uzupełnionego przedmiotami matematycznymi

Komitet C3S zaleca, by optymalny program studiów informatycznych zawierał łącznie 18 przedmiotów z dziedziny informatyki (wymienione wyżej przedmioty CS) i 7 przedmiotów uzupełniających — z dziedziny matematyki:
MA1 [7] — wstęp do rachunku różniczkowego i całkowego
MA2 — analiza matematyczna I
MA2a — teoria prawdopodobieństwa
MA3 — algebra liniowa
MA4 — struktury dyskretne
MA5 — analiza matematyczna II
MA6 — rachunek prawdopodobieństwa i statystyka matematyczna.

OBSŁUGA DYDAKTYCZNA KIERUNKÓW NIEINFORMATYCZNYCH

Odczuwa się wzrastającą potrzebę wprowadzania elementów informatyki do programów studiów na kierunkach nieinformatycznych. Zaleca się, aby przedmioty zawierające elementy informatyki wykładali przede wszystkim pracownicy wydziałów (instytutów) informatyki w ramach usług dydaktycznych. Obsługa dydaktyczna innych wydziałów poważnie zwiększa obciążenie wykładowców — informatyków. Kierując więc ich na te dodatkowe zajęcia trzeba dobrze rozważyć priorytety i możliwości kadrowe instytutu informatyki; nie można zapominać, że równocześnie należy zapewnić wysoki poziom kształcenia na kierunku „Informatyka".

Zakres tematyczny przedmiotów nauczanych w ramach usług zwykle jest bardzo zróżnicowany.

Podstawowe przedmioty nauczania w ramach usług

Studenci kierunków technicznych i matematycznych uzyskają najwięcej korzyści poznając zagadnienia objęte programem bazowym kierunku „Informatyka" (np. CS1). Jeżeli nawet student przerwie studiowanie (dodatkowo) przedmiotów informatycznych po drugim lub trzecim przedmiocie, to i tak wyniesie praktyczną znajomość co najmniej podstawowych technik programowania, które może wykorzystać w rozwiązywaniu zadań swojej dyscypliny. Dla studentów, którzy zechcą pogłębić wiedzę (na poziomie bardziej zaawansowanym niż CS1) zalecony jest zestaw przedmiotów wskazany w programie poziomu podstawowego.

Nauczanie informatyki na kierunkach nieinformatycznych obejmuje zazwyczaj: charakterystykę funkcjonowania komputerów, programowanie, zastosowanie wybranych metod i środków informatyki oraz efekty społeczne komputeryzacji. Proporcje i ujęcie poszczególnych zagadnień powinny być dostosowane do kierunku studiów, a stopień dostosowania winien być przedmiotem wspólnych ocen środowiska. Ze względu na tę różnorodność podano tylko jeden przykład: CSS1. Zastosowania komputerów i ich skutki (3-0-3). Program przedmiotu obejmuje zastosowania komputerów w takich dziedzinach, jak: utrzymanie zbiorów, gry decyzyjne, wspomagane nauczanie, sterowanie procesami, symulacja oraz modelowanie; wpływ komputeryzacji na jednostki i społeczeństwo; zastosowanie metod i środków informatyki w rozwiązywaniu problemów, z podkreśleniem analizowania, formułowania algorytmów oraz programowania. Ćwiczenia (projekty) powinny być zgodne z dziedziną zainteresowań studenta. Program przedmiotu obejmuje: systemy komputerowe, bazy danych, błędy programowania, wpływ komputeryzacji na społeczeństwo, skutki społeczne komputeryzacji w przyszłości, języki programowania, metody i techniki stosowane w rozwiązywaniu problemów, prace przejściowe i egzaminy. Udział każdego zagadnienia w programie nauczania przedmiotu zależy od specyficznych, lokalnych warunków.

Wspomagająca rola przedmiotów informatycznych

Mając możliwość wyboru, pewna liczba studentów obierze informatykę jako kierunek uzupełniający lub pomocniczy. Można wtedy zastosować wiele wariantów wyboru i łączenia poszczególnych przedmiotów informatycznych. Jedną z możliwości jest zaliczenie CS1 i CS2, a następnie co najmniej dwóch spośród CS3, CS4, CS5 oraz dwóch spośród CS6, CS7, CS8. Dodatkowe przedmioty mogą być dobierane zgodnie z wolą słuchacza i możliwościami uczelni.

Poniżej zestawiono przykładowo przedmioty informatyczne, które mają spełnić funkcję wspomagającą:
a) CS1 i CS2, jeden przedmiot obejmujący materiał CS5 i CS7 oraz jeden przedmiot obejmujący materiał CS3, CS4 i CS6
b) CS1 i CS2, jeden przedmiot obejmujący materiał CS3 i CS5 oraz jeden przedmiot obejmujący materiał CS4, CS6 i CS7.

W przypadku (a) w jednym bloku znalazły się podobne zagadnienia z różnych poziomów, natomiast podejście (b) polega na ujęciu różnych zagadnień na zbliżonym poziomie.

Jeżeli wykładane przedmioty informatyczne nie zadowalają słuchaczy, należy w miarę możliwości tworzyć programy i — zapewnić obsadę — nowych przedmiotów (najczęściej będą to przedmioty na wyższym poziomie).

Ustawiczne kształcenie

W sytuacji szybkiego rozwoju samej dyscypliny, jak i obszarów zastosowań nie jest możliwe formułowanie zawsze aktualnych zaleceń programowych. Niemniej wydziały (instytuty) informatyki powinny dostosować swoje możliwości do zmieniających się potrzeb. Zaleca się rozważenie następujących form podnoszenia poziomu wiedzy w dziedzinie informatyki:
— przedmioty poziomu zaawansowanego powinny być zbieżne z uprawianymi specjalizacjami uczelni

[7] MA to akronim ang. *mathematics* (matematyka)

— jak najczęściej należy organizować seminaria problemowe (np. minikomputery, systemy zarządzania bazami danych) — szkolenie może obejmować przedmioty typowe lub ich modyfikacje, może odbywać się w ramach zajęć wieczorowych lub w dni wolne od pracy (weekendy): przeznaczone jest głównie dla słuchaczy pracujących, pragnących zdobyć, odświeżyć lub pogłębić wiedzę z dziedziny informatyki.

WARUNKI REALIZACJI PROGRAMU

Realizacja idei nauczania informatyki zgodnie z intencją autorów Curriculum'78 wymaga nie tylko dostosowania programów do ich zaleceń. Niemniej ważne jest harmonijne dostosowanie programów kształcenia do programów szkół średnich, a także do wymagań pracodawców przyszłych absolwentów.

Podkreśla się różnorodność zasobów, jakimi dysponują uczelnie, w tym liczba etatów, wyposażenie laboratoryjne oraz poziom naukowy — co często związane jest z wielkością, lokalizacją, pozycją i przeznaczeniem zakładu naukowego oraz realizowanym programem kształcenia. Z tego powodu autorzy podali jedynie ogólne zalecenia co do warunków i sposobu realizacji programu, zdając sobie sprawę, że nie można sformułować zaleceń szczegółowych, jednakowo przydatnych w zróżnicowanych warunkach uczelni.

Program kształcenia

Według Komitetu C3S minimalny czas, jaki należy przeznaczyć na zajęcia w czasie jednego semestru, wynosi 36 godzin na przedmioty z dziedziny informatyki i 15 godzin matematyki. Jest to czas znacznie krótszy niż obecnie przeznacza się na realizację typowego programu studiów w naszym kraju.

Pożądane jest uzupełnianie zalecanego programu przedmiotami technicznymi, logiką, elementami fizyki, biologii itp., a w sferze zastosowań informatyki — przedmiotami ukierunkowującymi te zastosowania (np. zarządzanie). Niekiedy może to wymagać wprowadzenia dodatkowych przedmiotów już na poziomie podstawowym.

Autorzy raportu uważają, że absolwent kierunku „Informatyka" ze stopniem magistra z największym prawdopodobieństwem będzie zatrudniony początkowo jako programista (systemowy, naukowy, inżynieryjny lub ekonomiczny), a swoje programy będzie opracowywał przy pomocy specjalisty w określonej dziedzinie zastosowań. W ten sposób wiedza zdobyta na studiach — znajomość metod i środków informatyki — zostanie spożytkowana w praktyce do rozwiązywania specyficznych dla danej dziedziny zastosowań.

Wyposażenie laboratoryjne i biblioteczne

Podkreśla się, że informatyka jest w znacznym stopniu nauką empiryczną, podobnie jak biologia, fizyka i przedmioty techniczne. Nie wystarczy przygotowanie teoretyczne. Toteż realizacja zalecanego programu studiów wymaga odpowiedniego wyposażenia laboratoryjnego — niezbędne są urządzenia do wprowadzania danych, mikro- i minikomputery, duże i średnie systemy komputerowe, biblioteki algorytmów, a także języki programowania i struktury danych.

Postulat szerokiego udostępniania urządzeń laboratoryjnych w toku studiów może być spełniony w przypadku sprawnej współpracy instytutu informatyki, uczelnianego centrum obliczeniowego i innych instytutów korzystających z zasobów komputerowych uczelni.

W celu racjonalnego wykorzystania centralnie usytuowanych, dużych konfiguracji komputerowych, zaleca się powszechne stosowanie mikro- i minikomputerów. Użyteczność ich jest bardzo znaczna, umożliwiają one poznanie np. architektury komputerów (poprzez samodzielne montowanie mikrokomputerów zamiast symulacji i dyskusji na podstawie tekstów), umożliwiają bezpośrednią pracę studentów w trybie interakcyjnym przy użyciu języków nie stosowanych powszechnie w repertuarach maszyn średnich i dużych. Zaletą małych maszyn jest także niski koszt ich eksploatacji.

Studentom należy również zapewnić — niezależnie od możliwości korzystania ze zbiorów tradycyjnej biblioteki — dostęp do podręczników języków programowania i dokumentacji programów standardowych oraz do systemów komputerowych.

Kadra dydaktyczna

Konieczne jest, aby nauczyciele akademiccy na kierunku „Informatyka" — niezależnie od kierunku ukończonych studiów — mieli odpowiednie doświadczenie w stosowaniu metod i środków informatyki. Poza tym kwalifikacje należy stale doskonalić. Toteż kadra dydaktyczna powinna uczestniczyć w seminariach, w pracach na rzecz przemysłu.

Zależnie od wielkości wydziału różnej wysokości środki finansowe przeznacza się na kadrę dydaktyczną, której kwalifikacje decydują o poziomie i programie kształcenia. Trzeba to brać pod uwagę, gdyż realizacja uproszczonej wersji programu podstawowego — według zaleceń raportu — wymaga sześciu pełnoetatowych nauczycieli akademickich, przy założeniu, że większość z nich jest zdolna do nauczania dowolnego przedmiotu objętego programem podstawowym oraz niektórych przedmiotów dodatkowych, zgodnie z posiadaną specjalizacją.

Pracownicy ośrodka obliczeniowego, instytutu informatyki, matematyki i innych instytutów powinni zorganizować stałą pomoc dydaktyczną i metodyczną w nauczaniu przedmiotów związanych z informatyką i jej zastosowaniami. Poza kadrą etatową, wykładowcami pomocniczymi mogą być doświadczeni pracownicy administracji państwowej, przedsiębiorstw przemysłowych. Niektórzy z nich mogą wnieść odmienne spojrzenia na program kształcenia, należy jednak być ostrożnym w uzależnianiu realizacji programu od poglądów osób nie związanych na stałe z uczelnią.

Szybki rozwój dyscypliny wymaga ciągłych aktualizacji programu, modernizacji pomocy dydaktycznych i organizacji procesu dydaktycznego. Pociąga to dodatkowe obciążenia personelu dydaktycznego oraz pomocniczo-technicznego w wymiarze większym niż w bardziej ustabilizowanych dyscyplinach. Dlatego zaleca się, aby owe czynności organizacyjno-techniczne powierzyć wybranemu pracownikowi instytutu informatyki, który w zamian zobowiązany będzie do mniejszego pensum dydaktycznego.

Powyższy artykuł zawiera z konieczności bardzo ogólne tezy raportu Curriculum'78. Dlatego zainteresowanym przedstawicielom środowiska akademickiego gorąco polecamy oryginalny tekst raportu. Na szczególną uwagę zasługuje wiele wskazówek praktycznych dotyczących realizacji programu kształcenia informatyków w szkołach wyższych.

LITERATURA:

[1] Curriculum Committee on Computer Science (C3S). Curriculum'68, Recommendation for Academic Programs in Computer Science. Comm. ACM 11, 3 (March 1968)

[2] Curriculum'78. Recommendations for the Undergraduate Program in Computer Science. Comm. ACM, 22, 3 (March 1978)

[3] Austing R. H., Barnes B. H. and Engel G. L.: A Survey of the Literature in Computer Science Education since Curriculum'68, Comm. ACM 20, 1 (Jan. 1977)

[4] Education Committee (Model Curriculum Subcommittee) of the IEEE Computer Society. A Curriculum in Computer Science and Engineering. Committee Report, IEEE Pub. EHo 119-8, January 1977

[5] Committee on the Undergraduate Program in Mathematic. A General Curriculum in Mathematics for Colleges. Rep. to Math., Assoc. of America, CUMP, Berkeley, Calif., 1965

[6] Special Interest Group on Computer Science Education. SIGCSE Bulletin, ACM 5, 1 (Feb. 1973)

[7] Special Interest Group on Computer Science Education. SIGCSE Bulletin, ACM 6, 1 (Feb. 1974)

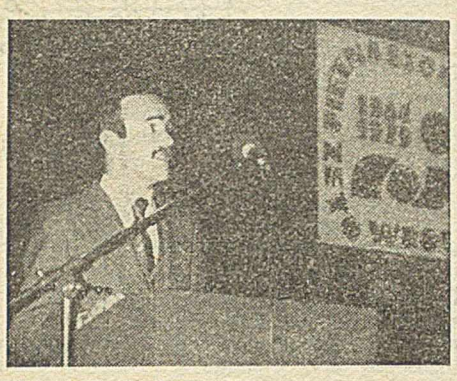
[8] Special Interest Group on Computer Science Education SIGCSE Bulletin, ACM 7, 1 (Feb. 1975)

[9] Special Interest Group on Computer Science Education. SIGCSE Bulletin, ACM 8, 1 (Feb. 1976)

[10] Special Interest Group on Computer Science Education. SIGCSE Bulletin, ACM 8, 3 (Aug. 1976)

Opracował J. BOBROWSKI

XV lat ZETO Wrocław



Przemawia dr J. Gwiazda

Jubileusze działalności informatycznej odliczamy ze zrozumiałych i powszechnie akceptowanych względów znacznie krótszymi czasokresami, niż to jest praktykowane w innych dziedzinach. Niezwykle szybkie tempo rozwoju informatyki i wynikające z niego zasadnicze przemiany formy i treści, a zwłaszcza gwałtowny wzrost rozmiarów działalności, wymagają tu częstszego niż gdzie indziej odnotowywania w kronikach historii.

Mimo więc, że od chwili, gdy rejestrowaliśmy X-lecie wrocławskiego ZETO *) upłynęło zaledwie 5 lat, pojawiła się znów nowa okrągła rocznica odmierzająca kolejny etap rozwoju tej niewątpliwie czołowej w skali kraju i szczególnie zasłużonej dla rozwoju polskiej informatyki placówki. Olbrzymi natłok w ubiegłym dziesięcioleciu faktów wynikających ze szczególnie dynamicznego rozwoju informatyki krajowej, nadrabiającej zapóźnienie lat sześćdziesiątych, w znacznym stopniu spowodował zatarcie właściwej hierarchii oceny poszczególnych osiągnięć, a zwłaszcza wkładu tych organizacji i zespołów ludzkich, które rozpoczynały działalność w okresie pionierskim.

Warto więc przypomnieć, że nasz obecny jubilat został powołany do życia 1 grudnia 1964 r. przez ówczesnego Pełnomocnika Rządu ds. Elektronicznej Techniki Obliczeniowej jako pierwsze ogniwo projektowanej krajowej sieci ogólnodostępnych ośrodków obliczeniowych. Decyzja lokalizacji tego ogniwa we Wrocławiu, jakkolwiek nieco zaskakująca dla najbardziej wtedy zaawansowanego i liczebnie największego środowiska stołecznego, nie była jednak przypadkowa. Szła ona bowiem konsekwentnie po linii zdyskontowania niezwykłej prężności środowiska wrocławskiego. Prężność ta, jak pamiętamy, pozwoliła już 5 lat wcześniej położyć w tym mieście podwaliny pod rozwój naszego przemysłu komputerowego.

*) patrz artykuł „ZETO Wrocław ma już 10 lat", INFORMATYKA nr 3/75, s. 38—40

Kolejne etapy coraz szybszego rozwoju ZETO Wrocław najlepiej można prześledzić poprzez rozwój środków technicznych, które ze zrozumiałych względów warunkowały rozmiary i zakres merytoryczny jego działalności.

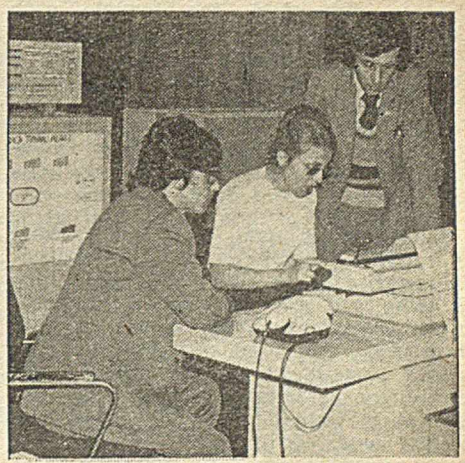
Okres pierwszy, to właściwie walka o uzyskanie możliwości rozwoju w okresie, jak pamiętamy, znacznego zwolnienia tempa rozwoju gospodarczego kraju, a także w otoczeniu społeczeństwa nie doceniającego jeszcze roli, jaką ma spełniać informatyka. W konsekwencji takiej sytuacji, niewielkie możliwości realizacyjne otrzymanego sprzętu komputerowego (ODRA 1003 w 1966 r. oraz MIŃSK 22 w 1967 r.), a także brak właściwych warunków lokalowych, nie pozwalają na szersze rozwinięcie statutowego zakresu działalności. Okres ten jednak nie tylko nie zniechęca kilkudziesięcioosobowego zespołu fanatyków informatyki, ale wręcz przeciwnie — pobudza do heroicznej walki o zdobycie własnego budynku. Wierzą oni, że nowa siedziba odpowiadająca najlepszym wzorom zagranicznym, nie tylko radykalnie poprawi ich pozycję w istniejącej strukturze społeczno-gospodarczej, ale przede wszystkim stworzy sprzyjające warunki do uzyskania właściwego sprzętu oraz bardziej wydajnej pracy szybko rosnącej załogi. I oto równo w 4,5 roku od daty powstania, w rekordowo krótkim cyklu inwestycyjnym ok. 2 lat, ZETO Wrocław uzyskuje nowoczesny obiekt, który przez kilka następnych lat będzie podstawowym wzorcem dla realizacji następnych nowoczesnych ośrodków obliczeniowych. Dążąc do możliwie najszybszego zapełnienia tego bardzo przestronnego wtedy obiektu sprzętem komputerowym wrocławianie podjęli się niezbyt wdzięcznej, a niedocenianej dotąd roli królika doświadczalnego Zakładów ELWRO. Rola ta dawała wprawdzie znaczne wyprzedzenie w otrzymaniu pierwszych egzemplarzy nowych konstrukcji ELWRO, ale pociągała za sobą przykre konsekwencje w postaci niewspółmiernie dużych kłopotów wynikających z użytkowej eksploatacji kolejnych prototypów (1969 r. — ODRA 1204, 1970 r. — ODRA 1304, 1973 r. — ODRA 1305).

Ewolucja sprzętu stymulowała oczywiście odpowiedni rozwój stopnia złożoności zastosowań realizowanych przez Zakład. Warto przy obecnej okazji przypomnieć, że symbolem postępu oraz osiągnięć znacznie wyprzedzających ówczesny średni poziom krajowy i stąd silnie akcentowanych z okazji jubileuszu X-lecia, było opracowanie, uruchomienie i rozpoczęcie eksploatacji wielodostępnego systemu informatycznego POLRAX-2, zrealizowanego w zasadniczej części na sprzęcie krajowym.

Etap ostatniego pięciolecia działalności ZETO Wrocław należy mierzyć już inną skalą osiągnięć, charakterystycznych dla współczesnego okresu rozwoju informatyki polskiej. Instalowanie nawet bardzo rozbudowanych konfiguracji systemów komputerowych, czy też oddawanie do użytku coraz bardziej obszernych i funkcjonalnych budynków ośrodków informatycznych stało się już zjawiskiem codziennym, coraz rzadziej odnotowywanym w kronice wydarzeń krajowych, nawet na łamach INFORMATYKI.

Najistotniejszą sprawą stało się dążenie zarówno do pełniejszego wykorzystania coraz bardziej kosztownego sprzętu, jak i wzrostu efektywności realizowanych zastosowań. I w tej dziedzinie ZETO Wrocław stara się utrzymać pozycję czołową jako największe przedsiębiorstwo sieci ZETO (ok. 1300 pracowników, ponad 1/5 łącznej wartości sprzedanych usług). Oprócz 5 oddziałów i ośrodków zamiejscowych (Opole, Wałbrzych, Zielona Góra, Oława, Leszno Wielkopolskie), Wrocław zorganizował samodzielną już Jelenią Górę, a wiele innych placówek, nie tylko zresztą sieci ZETO, bezpośrednio korzystało z jego pomocy organizacyjnej i doświadczeń.

ZETO Wrocław dysponuje obecnie nie tylko największym w sieci parkiem komputerowym, ale również oferuje najszerszy zakres usług. Oprócz bowiem usług typowych (projektowanie, programowanie, przygotowanie maszynowych nośników informacji, przetwarzanie, szkolenie użytkowników) rozwinięto również takie usługi, jak serwis techniczny komputerów ODRA 1300, minikomputerów RC i DP, urządzeń zdalnego dostępu, urządzeń do przygotowania maszynowych nośników infor-



Początki eksploatacji POLRAX-2

macji ARITMA i urządzeń klimatyzacyjnych OKG, a także projektowanie organizacji i technologii ośrodków obliczeniowych, projektowanie systemów cyfrowych oraz regeneracja magnetycznych nośników informacji.

Imponujące są również dotychczasowe wyniki szkolenia użytkowników — w ciągu całego okresu działalności zdobyło tu nowe kwalifikacje zawodowe w trybie szkolenia kursowego ponad 7200 osób, a więc ok. 15% łącznej statystycznej liczby osób czynnych zawodowo w informatyce krajowej.

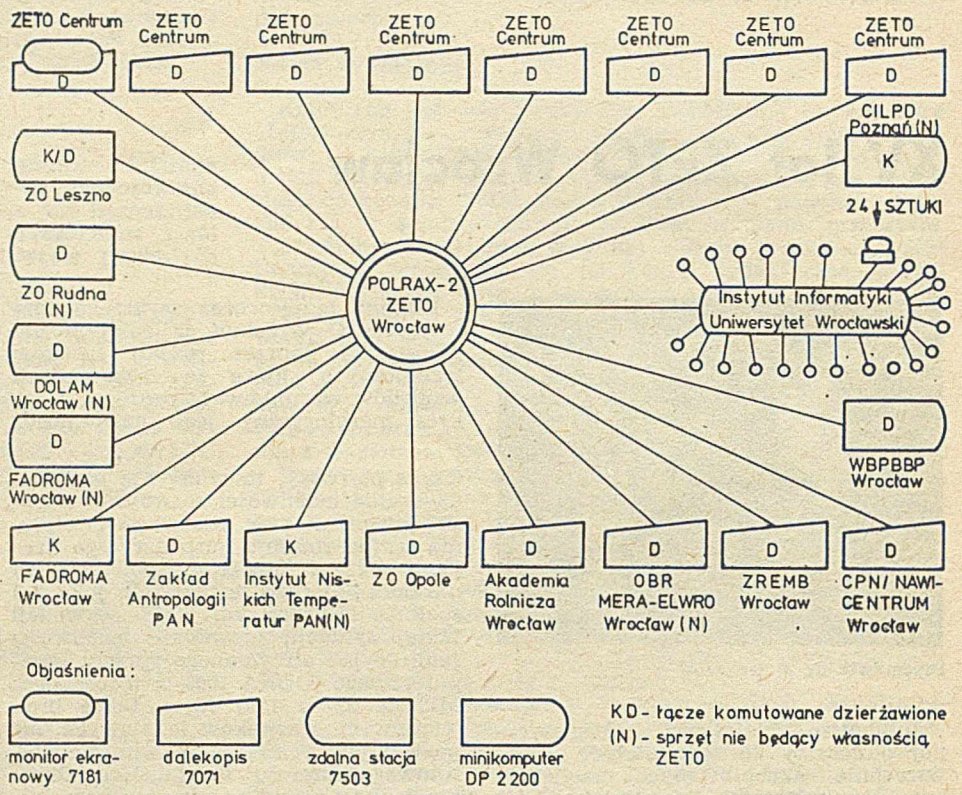
Większość obecnych usługobiorców to jednostki gospodarcze z terenu 5 sąsiadujących województw, należące głównie do resortów: przemysłu maszynowego, hutnictwa, przemysłu maszyn ciężkich i rolniczych, rolnictwa, przemysłu chemicznego i finansów. Eksploatowany sprzęt charakteryzuje się znaczną jednolitością (4 zestawy typu ODRA 1304 oraz 6 zestawów typu ODRA 1305), uzupełnioną egzemplarzem R-32. Od 1974 r. park ten uległ wprawdzie w wyniku zmian organizacyjnych ilościowemu zmniejszeniu o ok. 15% (z 13 do 11 zestawów), lecz jego wartość w ujęciu pieniężnym została podwojona (z 201 do 408 mln zł), co najlepiej wyjaśnia obecny potencjał Wrocławia.

Szczególnie interesujące ze względu na możliwość syntetycznej oceny działalności ZETO Wrocław są oczywiście wskaźniki wydajności. W omawianym okresie 5 ubiegłych lat obroty wzrosły więcej niż dwukrotnie (ze 189 do ok. 400 mln zł), natomiast zatrudnienie zaledwie o ok. 10% (z 1165 do 1284 pracowników). Daje to prawie dwukrotny wzrost wydajności na 1 zatrudnionego (z ok. 162 do ok. 311 tys. zł).

Najbardziej istotną pozycją wśród osiągnięć ostatnich lat działalności ZETO Wrocław był niewątpliwie rozwój wspomnianego systemu POLRAX-2. Jego obecny, imponujący zakres funkcjonowania ilustruje zamieszczony schemat. System ten stanowi niezaprzeczalny wkład w krajowy rozwój postępowych form realizacji zastosowań informatyki, inspirując m.in. odpowiednie ukierunkowanie ewolucji sprzętu przez nasz przemysł komputerowy.

Na marginesie tej ostatniej sprawy należy stwierdzić, że niewiele osób zdaje sobie sprawę z tego, jak istotny był wkład wrocławskiego ZETO w osiągnięcie obecnego poziomu rozwoju sprzętu krajowego w wyniku wieloletniego pełnienia wspomnianej już roli „królika doświadczalnego" w odniesieniu do prototypowych rozwiązań konstrukcyjnych.

Dlatego też nasz jubilat jest wart nie tylko opublikowania najserdeczniejszej choćby informacji okolicznościowej, ale również bardziej konkretnych wyrazów uznania. Sądzę, że najbardziej pożądanym, a jednocześnie w pełni uzasadnionym potwierdzeniem tego uznania byłaby decyzja realizacji zaprojektowanej już przed kilku laty nowej siedziby przedsiębiorstwa. Opisany rozwój działalności spowodował w ostatnich latach znaczne przekroczenie



Obecny stan rozwoju systemu POLRAX-2

dopuszczalnych granic zagęszczenia sprzętem i ludźmi dotychczas posiadanych powierzchni, mimo ich wielokrotnego rozszerzania wynajmem dodatkowych pomieszczeń w różnych częściach miasta.

Budowę nowej siedziby ZETO Wrocław należy jednak rozpatrywać nie tylko w kategoriach bezpośrednich potrzeb przedsiębiorstwa, ale również jako niezbędny warunek bardziej sprawnego zaspakajania potrzeb obliczeniowych dotychczasowych, a także coraz liczniejszych nowych użytkowników. Oznacza to niezbędne hamowanie naturalnych, lecz w wielu przypadkach całkowicie nieuzasadnionych dążeń do tworzenia przez tych użytkowników własnych ośrodków obliczeniowych, a więc niepożądanego — zwłaszcza w obecnej sytuacji — rozpraszania środków inwestycyjnych. Instancje odpowiedzialne za prawidłową realizację koordynacji terenowej powinny argumenty te wnikliwie przeanalizować przy podejmowaniu przyszłych decyzji inwestycyjnych informatyki na terenie Wrocławia. **(W.K.)**

Uroczystość jubileuszowa

Dotychczasowe osiągnięcia ZETO Wrocław i jego wysoka ranga na tle całokształtu polskiej informatyki zostały uhonorowane uroczystą akademią. W dniu 6 grudnia ub.r., w trakcie jubileuszowego spotkania załogi z przedstawicielami władz i licznej grupy użytkowników, dyrektor naczelny mgr Dariusz Ozobiński przypomniał heroiczny okres powstawania Zakładu oraz kolejne etapy jego rozwoju.

Zaangażowanie pracowników przedsiębiorstwa w proces unowocześniania gospodarki regionu znalazło swój wyraz w przyznaniu im wielu odznaczeń państwowych, regionalnych, branżowych i związkowych. Najwyższe odznaczenie — Krzyż Kawalerski Orderu Odrodzenia Polski — otrzymał szczególnie zasłużony, a jednocześnie jeden z pierwszych pracowników tej placówki, mgr Andrzej Ramułt. Z okazji jubileuszu przyznano też po raz pierwszy odznaki „Zasłużony dla ZETO".

Dużą satysfakcję przyniosły załodze wystąpienia okolicznościowe dyrektora Departamentu Informatyki MNSzWiT dr Janusza Gwiazdy (w imieniu kierownictwa resortu), a także dyrektora naczelnego Zjednoczenia Informatyki mgr Zbigniewa Substyka. Nie zabrakło również gratulacji od licznych stałych klientów, których łączą z ZETO Wrocław szczególnie ścisłe więzy. Cała bowiem trudna droga rozwoju, od początków powstania Zakładu aż po dzień dzisiejszy, była pokonywana wspólnie z użytkownikami.

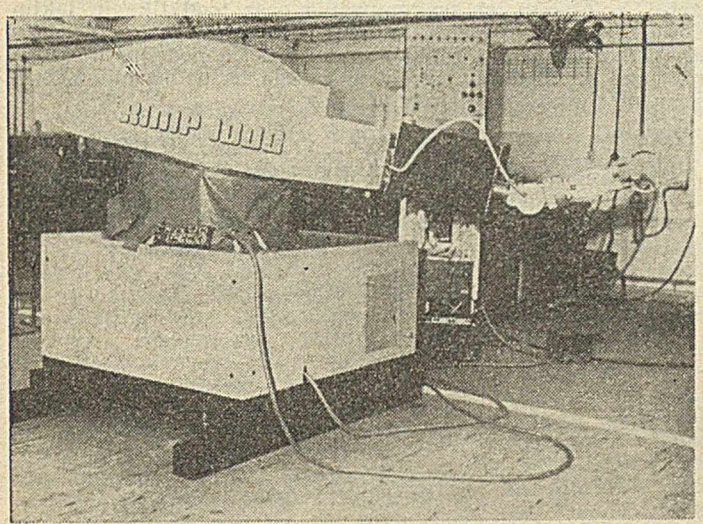
Do licznych, w pełni zasłużonych gratulacji i życzeń dalszych sukcesów przyłącza się również cały zespół redakcyjny INFORMATYKI. **(K.B.)**

System CAMAC w sterowaniu robotem przemysłowym

W końcu 1977 roku w Instytucie Mechaniki Precyzyjnej powstał robot przemysłowy RIMP 1000, oparty na amerykańskiej konstrukcji Unimate 2000B. Robot przeznaczony jest do wykonywania operacji technologicznych wymagających wysokiej precyzji i powtarzalności, takich jak: zgrzewanie blach, spawanie łukowe, obsługa wtryskarek i pras, prace załadunkowe i wyładunkowe, prosty montaż. Jest to robot pierwszej generacji, o manipulatorze pracującym w biegunowym układzie współrzędnych. Napęd hydrauliczny z wykorzystaniem serwozaworów pozwala na ruchy z zadanymi prędkościami i umożliwia zatrzymanie głowicy robota w uprzednio zaprogramowanym miejscu.

Układ sterowania robotem rozwiązano w oparciu o system CAMAC *(Computer Application for Measurement And Control)* i użycie autonomicznego procesora 131. Pozwoliło to na wykorzystanie bloków systemowych produkowanych w kraju oraz istniejących opracowań dotyczących modułów CAMAC-a. Zastosowano tu uniwersalny zestaw do pracy w czasie rzeczywistym, oparty na procesorze 131, pamięciach danych i programu oraz standardowych interfejsach (do drukarki, czytnika i pamięci kasetowej). Układ sterowania zawiera ponadto szereg bloków specjalistycznych, opracowanych w Instytucie i wykonanych w standardzie CAMAC lub też korzystających ze wspólnej kasety, ale nie mających połączenia z magistralą. Funkcje układu sterowania są następujące:

- sterowanie wykonywaniem sześciu niezależnych ruchów (6 stopni swobody) z programowaną prędkością
- pozycjonowanie typu PTP *(point to point)* z założoną dokładnością maks. 1,5 mm, wynikającą ze zdolności rozdzielczej enkoderów [1]
- możliwości pracy w następujących trybach: a) uczenie, b) odtwarzanie, c) zapis do pamięci kasetowej, d) odczyt z pamięci kasetowej, e) praca ciągła — dynamiczna współpraca z pamięcią kasetową, f) współpraca z zewnętrznym systemem komputerowym, g) testowanie.



Robot przemysłowy RIMP 1000 przeznaczony jest do wykonywania operacji technologicznych wymagających wysokiej precyzji i powtarzalności

[1]) Enkoder to urządzenie mechaniczno-elektroniczne zamieniające kąt obrotu wałka (położenie) na wartość cyfrową; odczyt pozycji uzyskuje się na wyjściu fotodiod oświetlanych światłem przechodzącym przez tarczę kodową osadzoną na osi obrotu wałka

Do pamiętania parametrów ruchu wykorzystano organizację kroku składającego się z 8 słów 24-bitowych; dwóch funkcyjnych i sześciu określających wartość przemieszczenia jednostek napędowych poszczególnych stopni swobody. Słowa funkcyjne zawierają informację dotyczącą wyboru prędkości ruchu (jednej z ośmiu), programowanego opóźnienia (0—99 s), dokładności pozycjonowania (5 możliwości), sterowania maksymalnie 9 urządzeniami zewnętrznymi oraz ewentualnym zaciskaniem szczęk chwytaka. Liczba kroków odpowiadająca stosowanej pamięci wynosi 128 lub 512. Ograniczenie to wynika z pojemności obecnie produkowanych w kraju pamięci do systemów CAMAC. W przypadku zbyt małej pojemności pamięci operacyjnej można wykorzystać pamięć masową (bęben, dyski, taśmę) i traktować ją jako bibliotekę typowych programów.

Program sterujący może zajmować część pamięci danych lub może być oddzielnym blokiem pamięci stałej. Zaletą przechowywania programu w pamięci stałej jest wyeliminowanie konieczności wprowadzania programu do pamięci operacyjnej przy każdym rozruchu. Pamięć danych może być podzielona na 8 niezależnych podprogramów. Oznacza to, że każdy podprogram wprowadzony przy uczeniu robota stanowi odrębny fragment i może być odtworzony jako dowolna kombinacja podprogramów. Układ prowadzi kontrolę poprawności zapisywanej i odczytywanej informacji. Każde słowo odczytane z bloku wejściowego jest uzupełnione kontrolnym bitem parzystości. Przy wpisywaniu do pamięci sprawdzana jest poprawność zapisu przez porównanie informacji zapisanej z informacją w rejestrach bloku wejść. W trakcie odczytywania kontrolowana jest parzystość kontrolowana (przesłaniem słowa do rejestrów sprawdzających lub programowo). Układ może sterować i kontrolować pracą 9 urządzeń zewnętrznych przez synchronizację pracy robota z sygnałami potwierdzeń przychodzącymi z urządzeń współpracujących. Synchronizacja pracy linii technologicznej z użyciem robotów stanowi osobny problem, niemniej istnieje techniczne sprzężenie z zewnętrznym systemem komputerowym.

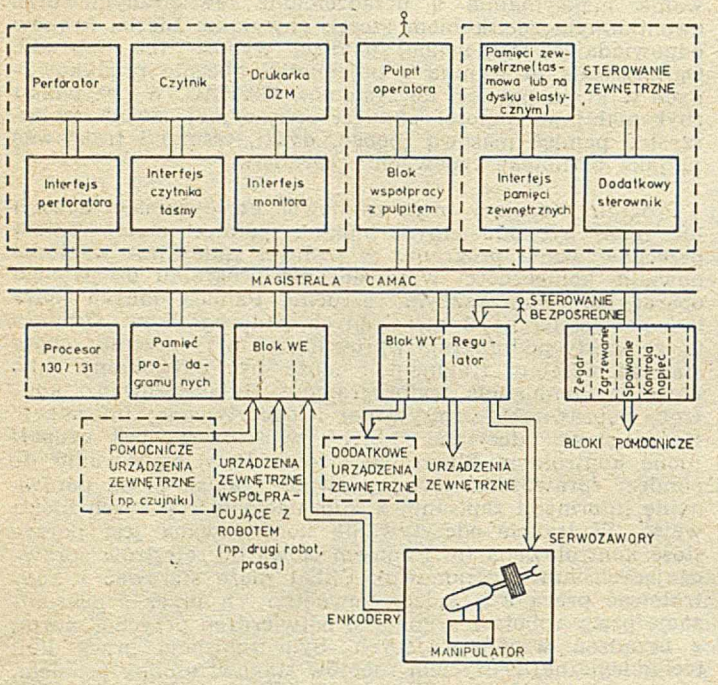
Ogólna zasada pracy układu sterowania polega na kształtowaniu odpowiedniego przebiegu napięcia, podawanego na serwozawory bezpośrednio sterujące ruchami jednostek napędowych. Algorytm pracy robota jest realizowany na podstawie porównania aktualnego położenia jednostki ruchu z położeniem zadanym; układ otrzymuje informację o aktualnym położeniu z przetworników kątowych (enkoderów) pracujących w ujemnym kodzie Graya.

Układ sterowania robotem RIMP 1000 składa się z jednostki centralnej pamięci, bloków pośredniczących: wejściowego, wyjściowego i współpracy z konsolą oraz bloków pomocniczych, które kontrolują przebiegi specyficznych procesów (zgrzewanie, spawanie), odmierzają czas, sprawdzają napięcia.

Oprogramowanie podstawowe systemu bazuje na następującym algorytmie. W trybie pracy „uczenie" do pamięci operacyjnej zostają wprowadzone dane zgodnie z podaną uprzednio strukturą kroku. Położenie określone w 14 bitach pochodzących z enkoderów zostaje zamienione w bloku wejściowym na kod binarny i uzupełnione bitem parzystości. Przy odtwarzaniu aktualny stan enkodera jest próbkowany co 2 s, odejmowany w procesorze od stanu pobranego z pamięci, a otrzymana różnica jest przesyłana do regulatora, który na jej podstawie wytwarza przebieg napięcia realizujący algorytm ruchu danego członu napędowego. Regulator zapewnia bezprzeciążeniowy ruch (ograniczenia konstrukcyjne dopuszczają przeciążenia do 2 g) oraz pozycjonowanie z dokładnością ±1,5 mm. W czasie, gdy w regulatorze następuje ładowanie pamięci analogowej (ok. 200 μs) próbkuje się stan następnego enkodera. Cykl zostaje zakończony po stwierdzeniu dojścia wszystkich jednostek napędowych do zadanych położeń. Stan zgodności ogólnej zezwala na przejście do następnego kroku. Procesor 131, stanowiący jednostkę centralną systemu, umożliwia pracę krok po kroku oraz pojedynczymi cyklami. Po zatrzymaniu

robota można także przejść do dowolnie wybranego kroku. Pozwala to na korekcję pojedynczych kroków i odtwarzanie wybranych fragmentów programu.

Komunikacja systemu z operatorem odbywa się przez blok współpracy z konsolą. Pojawienie się dowolnego przerwania, które spowodowane jest akcją operatora, powoduje wpisanie danego przerwania do rejestru stanu i wygenerowanie zgłoszenia bloku. Po rozpoznaniu zgłoszenia zostaje pobrane słowo stanu. Obsłużone zostaje przerwanie o największym priorytecie i zgaszony bit przerwania w słowie stanu.



Schemat blokowy układu sterowania (linią przerywaną oznaczono bloki dodatkowe układu podstawowego)

Dysponując opisaną konfiguracją układu sterowania i manipulatorem, przez dołączenie nowych bloków można znacznie rozwinąć system. Możliwości rozwojowe obejmują wprowadzenie rozejść warunkowych polegających na przejściu do podprogramów pod wpływem zewnętrznych sygnałów. Można to wykorzystać do rozdzielania elementów typów A i B pod warunkiem uzyskania rozróżnialności elementów np. przez wprowadzenie „czucia". Korzystając z możliwości obliczeniowych procesora można uzyskać liniowe przejście głowicy między dwoma zadanymi punktami, natomiast punkty pośrednie uzyskuje się przez przeliczenie współrzędnych biegunowych mierzonych w układzie robota na bieżące współrzędne w układzie prostokątnym.

Kolejnym udogodnieniem może być wprowadzenie tzw. kroków bezpiecznych. Krokiem bezpiecznym nazywamy taki krok, po którym można wykonać powrót do punktu początkowego najkrótszą drogą. Ma to istotne znaczenie przy pracy w otoczeniu „przeszkód", kiedy niemożliwy jest powrót do punktu początkowego z pozycji, w jakiej robot został wyłączony. W takim przypadku można doprowadzić robota do wymaganej pozycji ręcznie albo po włączeniu pracy automatycznej. Można też odtworzyć stan sprzed wyłączenia i kontynuować zaprogramowany ruch aż do napotkania kroku bezpiecznego lub końca programu.

Zwiększenie możliwości systemowych wiąże się z powiększeniem wielkości programu, pojemności pamięci danych (ewentualnie dołączeniem pamięci zewnętrznych) oraz zwolnieniem czasu reakcji przy skomplikowanych zadaniach. Problem pojemności pamięci został już rozwiązany z chwilą wprowadzenia na rynek dużych pamięci półprzewodnikowych. Obecnie największym kłopotem jest znalezienie właściwych metod analizy, zaprojektowanie „inteligentnych" urządzeń zewnętrznych współpracujących z robotem oraz stworzenie języka konwersacyjnego człowiek—system dla pełnego wykorzystania autonomicznego systemu robota przemysłowego.

Aktualnie w Instytucie Mechaniki Przemysłowej prowadzi się prace nad:

- udoskonaleniem układu sterowania przez
  — budowę układu sterowania w oparciu o elementy wielkiej skali inteligencji z rodziny MCS 80
  — budowę modułowego układu sterowania z różnymi blokami wykonawczymi; oprócz odtwarzania pozycji, tak jak w sterowaniu CP *(Continuous Path)*
- zwiększeniem niezawodności pracy
- zastosowaniem robota i linii robotów w przemyśle; nowe zadania to sterowanie nadrzędne, uproszczenie programowania robota oraz zapewnienie 100% bezpieczeństwa pracy.

W najbliższej przyszłości przystąpi się do opracowania prostszego sposobu uczenia robota przez opisywanie czynności, kolejności ich wykonywania oraz modyfikacji, jakim dana czynność ma podlegać. Jest to szczególnie istotne przy długich programach z powtarzającymi się podobnymi czynnościami (np. rozładunek palety polega na pobieraniu takich samych elementów z różnych miejsc).

Andrzej GOGOLEWSKI
Instytut Mechaniki Precyzyjnej
Adam ŻUKOWSKI
Politechnika Warszawska

Czytelniku,

czy pamiętasz o prenumeracie INFORMATYKI?

Przypominamy, że 10 marca upływa

termin przyjmowania zamówień na II kwartał 1980 r.

Odrowski remanent

Do pewnego lorda angielskiego, a było to przed wielu laty, przyjechał z wizytą kolega z lat szkolnych.

— Słyniesz na równi z bogactwa jak i z rozrzutności — mówi — czy możesz mi wytłumaczyć na czym to polega?

— To bardzo proste — pada odpowiedź — obejrzyj jednak moją stajnię i garaże.

— Sądzę, że tego właśnie araba dosiadasz najczęściej — pyta gość.

— Nie, kupiłem go, bo był drogi i ze szlachetnym rodowodem. Raz użyłem go do polowania.

— A zapewne ten rolls-royce jest twym ulubionym pojazdem?

— Byłem nim kiedyś na weekendzie, ale teraz po prostu cieszy moje oko.

Udają się na lunch. Lord przedstawia swą córkę.

— Oczywiście jedynaczka? — przytomnie pyta kolega.

Bardzo proszę uwierzyć, że nie tylko kokieteria wobec czytelników podyktowała takie właśnie wprowadzenie. Anegdota powyższa nieomal natrętnie kojarzy się z tematem, o którym tu pisać zamierzam: o komputerach ODRA 1305. I o organizacji ETOB, która nie chciałaby się z owym lordem identyfikować.

Sieć ETOB eksploatuje obecnie 19 komputerów ODRA 1305. Nieomal wyłącznie na trzy zmiany. Aż dziesięć z nich ma pamięci operacyjne o pojemności 64 K, pięć — 96 K, a cztery — 128 K. Tylko dwie jednostki są wyposażone w pamięci dyskowe (ICL) o pojemności 60 MB. Dziesięć dysponuje 8 MB, bułgarskiej głównie produkcji. Aż siedem więc to komputery w konfiguracji wyłącznie taśmowej, przy czym też siedem (choć nie dokładnie te same) ma mniej niż dwanaście jednostek PT-3.

Drukarek DW-325 jest zaledwie 26, przy czym tylko siedem komputerów ma dwie drukarki szybkie. W ośmiu przypadkach niedostatek tych urządzeń jest uzupełniany wyrobem o symbolu DW-312. Pozostałe komputery nawet takiej asekuracji nie mają.

Udało się nam niedawno zakupić 34 lokalne monitory ekranowe VIDEOTON, z dziewięcioma jednostkami sterującymi. Warunkiem jednak efektywnego wykorzystania tych urządzeń jest stosowanie systemu operacyjnego GEORGE 3. A wiadomo, że progiem racjonalnego stosowania tego systemu są pamięć operacyjna 96 K (powinno być 128 K), niezawodne pamięci dyskowe — czego o bułgarskich powiedzieć się nie da — oraz trzy szybkie drukarki wierszowe.

Od 1979 roku sieć ETOB posiada cztery multipleksory MPX-325. Towarzyszy im jednak zaledwie 21 najmniej „inteligentnych" terminali w postaci DZM 180 KSRE. Dobre to dla trenowania teletransmisji, ale dalece nie wystarczające dla znaczącego użytkowo jej stosowania.

W 1980 roku organizacja ETOB będzie się składała z 18 wyposażonych w komputery usługowych ośrodków obliczeniowych, co na dłuższy czas powinno zaspokoić potrzeby rozwojowe sieci informatycznej budownictwa. Program rozwoju ekstensywnego jest zatem bardzo skromny. Tym większy nacisk musimy położyć na rozwój zwany intensywnym. A tu nie ma wyboru: docelowo jeszcze tylko 2—4 nowe ODRY 1305, ale przede wszystkim racjonalne uzupełnianie ich konfiguracji. I dominacja tych właśnie maszyn w najbliższym dziesięcioleciu. Jest to po prostu konsekwencja orientacji na sprzęt krajowy, zdecydowanej — po myśli producenta komputerów — przed paroma laty. Na te właśnie maszyny mamy stosunkowo bogate i stale rozwijane oprogramowanie użytkowe, wyspecjalizowaną kadrę projektową i eksploatacyjną. Mamy też własny, wypróbowany serwis techniczny, co jest wysoce wygodne zwłaszcza dla producentów sprzętu.

Zaczęliśmy — owszem — kupować również maszyny R-32, w pełni świadomi, że górują one nad ODRAMI głównie konfiguracją dostarczanych zestawów. Ważny to wprawdzie argument, ale z wielu względów — w których dominuje potrzeba jednolitości i uniwersalności usług dla budownictwa — nie przewidujemy, aby do końca 1982 r. więcej niż trzy z osiemnastu ośrodków mogły się obejść bez komputerów ODRA 1305. Okazuje się więc — a nie jest to odkrywcze stwierdzenie — że łatwiej o koniunkturalne zmiany produkcji sprzętu niż o rozsądne, ekonomicznie użytkowe za tymi zmianami nadążanie.

Wracając do wątku głównego, nie zaszkodzi powtórzyć, że około dziesięcioletnia perspektywa eksploatowania ponad dwudziestu maszyn ODRA 1305 jest kanwą tego artykułu. Warto więc poddać w wątpliwość określenie komputera średniej mocy w odniesieniu do urządzenia, które w latach osiemdziesiątych dysponuje pamięcią operacyjną o pojemności 64 czy nawet 96 K. A przecież przy pełnym wyposażeniu pamięciowym i peryferyjnym ODRA 1305 przez ładnych jeszcze parę lat skutecznie wytrzymałaby rywalizację z wiadomymi konkurentami. Zwłaszcza przy uwzględnieniu stanu oprogramowania — tak podstawowego, jak użytkowego.

Konkretyzując sprawę, należałoby zapytać decydentów i dystrybutorów maszyn ODRA 1305, czy mogą spać spokojnie, wiedząc, iż miliardy, jakie włożyliśmy w licencję i produkcję tego sprzętu, tak owocują, że dobre i stosunkowo szybkie jednostki centralne tracą głównie czas w oczekiwaniu na możliwość wydruku na jednej przeważnie drukarce wierszowej? Dodanie drugiej drukarki, za cenę ok. jednej dziesiątej kosztów zestawu, zwiększa wydajność przetwarzania niemal dwukrotnie. Ale kupić trudno, choć mamy bodaj największą w Europie fabrykę takich drukarek.

Pamięci dyskowych nie produkujemy, bo tak z podziału pracy wynikło. A nawet gdyby te dyski były niezawodne i łatwo dostępne, to i tak są one o rząd pojemności za małe, aby efektywnie wykorzystać możliwości eksploatacyjne jednostek centralnych ODRY 1305 oraz możliwości użytkowe oprogramowania podstawowego tych maszyn.

Jest sprawą oczywistą, że niniejsza wypowiedź, mimo iż nacechowna gospodarską troską, zawiera pewien dysonans wobec propagowanej obecnie polityki rozwoju produkcji sprzętu informatycznego. Nie potrafię znaleźć rzeczowego kontaktu z realiami tej polityki. Jestem natomiast skłonny — choćby ze względu na szacunek do adwersarzy — wczuć się w ewentualne intencje krytyki etobowskiego stanowiska.

Nie rozmieniając polemiki na drobne, postawiłbym etobowcom jeden tylko zarzut. Jeśli bowiem naprawdę chcieliby pełnego pod względem użytkowym uzupełnienia konfiguracji maszyn ODRA 1305, to rychło by się okazało, że ich realny potencjał komputerowy uległby co najmniej pięciokrotnemu zwiększeniu wydajności w przetwarzaniu danych, a tego sieć ETOB nie potrafiłaby zagospodarować, co wspomnianą na wstępie lordowską rozrzutność mogłoby na myśl przywodzić. I byłaby to prawda, lecz tylko teoretyczna, niestety.

Począwszy od 1979 roku, ETOB mimo rosnącego popytu jest zmuszony do konsekwentnego odmawiania świadczenia nowych usług informatycznych. Dotyczy to spraw tak dla budownictwa newralgicznych, jak służby finansowo-księgowe, obsługa fabryk domów, wykorzystanie ciężkiego sprzętu budowlanego, obliczanie płac itp. I to nie z powodu braku komputerów, lecz na skutek drastycznie ubogiego ich wyposażenia w pamięci operacyjne i urządzenia peryferyjne.

Szczególnie dramatyczny niedosyt występuje w jednostanowiskowych urządzeniach do tworzenia nośników danych w miejscach ich powstawania. Była o tym mowa w nr 11/79 INFORMATYKI. Aktualne potrzeby resortu budownictwa to kilkaset urządzeń typu MERA 301, MERA 100 lub PSPD-90, z odpowiednią liczbą konwerterów. Jednak na zasadach nie polegających na zaliczaniu producentowi ich kupna do sprzedaży eksportowej.

Niemniej ważną sprawą jest szybkie rozwijanie usług opartych na teletransmisji. Czeka na nie blisko 150 fabryk domów, ponad sto biur projektów oraz kilkaset przedsiębiorstw budownictwa, którym coraz bardziej nie wystarcza dekadowy i miesięczny cykl świadczenia usług informatycznych. A konfiguracje wyposażenia komputerów oraz oferowane dostawy MPX-325 + 4 sztuki DZM 180 KSRE w tym kontekście wyglądają na ponury żart.

Nie mam złudzeń, że powyższy tekst może poruszyć czyjeś sumienie. Niech więc będzie choćby świadectwem rzeczywistości, aby na przykład pointę wstępnej anegdoty ze zrozumieniem do organizacji ETOB odnosić.

Wincenty ŁADA

Gdzie diabeł nie może...

Do portretu pozują (od lewej): inż. Mieczysława Klochowicz, mgr inż. Wacława Michałowska, mgr Krystyna Dobrowolska, mgr Izabela Gruszecka, Agnieszka Anacka, mgr Halina Brzezińska, Irena Piechota, Barbara Popławska, mgr Anna Zagalska, mgr Jolanta Nagórska, Ewa Wiatrowska, Julita Gozdan

Fot. T. Zagoździński

Gdyby dosłownie rzecz traktować, to bohaterem tytułowym byłby doc. dr Czesław Uhma, zastępca dyrektora Centralnego Ośrodka Badawczo-Rozwojowego Przemysłu Betonów „CEBET", który przed ośmiu laty zainspirował utworzenie w Centrum ETOB informatycznego zespołu projektowo-programowego dla przemysłu betonów. I wcale mu nie przeszkadzało, że branża — męskiej raczej ręki wymagająca — została w informatyce obsadzona zespołem całkowicie kobiecym. A chodziło o zespół nieomal „do wszystkiego" — od projektowania systemów do kruszenia oporu użytkowników w fazie eksploatacji. Tak więc tytuł tego portretu istotę rzeczy wiernie oddaje.

Zjednoczenie Przemysłu Betonów zarządza 27 przedsiębiorstwami, z których wiele ma bogate tradycje, bo jeszcze z okresu manufaktur się wywodzące. Radykalne unowocześnienie technologii stało się faktem, ale nie bez trudu i oporów. A wiadomo, że postęp organizacyjny jeszcze trudniej się wprowadza i jest on zwykle wobec postępu technologii opóźniony. I właśnie informatycy konsekwencjami takiego stanu rzeczy są szczególnie doświadczani.

Zespół, o którym mowa, otrzymał na początku wielce obiecujące zlecenie generalne: opracować w miarę uniwersalny informatyczny system zarządzania dla przedsiębiorstw podległych zjednoczeniu. System opracowano, podjęto jego wdrażanie — poprzedzone zorganizowaniem i instruktażem zakładowej komórki ETO. Zdarzyło się jednak, że w tym właśnie czasie zmieniono dyrektora przedsiębiorstwa. Nowy dyrektor już w pierwszych dniach urzędowania zlikwidował z trudem utworzoną komórkę ETO i odrzucił system informatyczny, słusznie poniekąd sądząc, iż to on, a nie komputer, będzie zarządzał przedsiębiorstwem.

Cierpliwy „CEBET" znalazł innego ochotnika — w Ostrowie Wielkopolskim — wyciągając wraz z etobowskim paniami wnioski praktyczne na temat organizacji wewnątrzzakładowych służb ETO i sposobów zintegrowania dyrekcji. Wpajano m.in. poczucie odpowiedzialności za pilotowość przedsięwzięć w skali zjednoczenia. Stworzono atmosferę rywalizacji między przedsiębiorstwami w zastosowaniach informatyki. Różne były tego skutki. M.in. takie, że deklarowali się dyrektorzy, a odpowiedzialni za sprawę pracownicy dokładali niekiedy starań, aby udowodnić, że zamysły informatyzacyjne są niecelowe. Stało się więc jasne, że bez zaangażowanej komórki, czy choćby jednej osoby w przedsiębiorstwie, trudno liczyć na pozytywne wyniki.

Pojawiły się i wyniki. Ledwie wdrożono system kontroli realizacji dostaw, okazało się, że od zaraz potrzebny jest system planowania produkcji w powiązaniu z dostawami. Różnie go zresztą potem traktowano, ale o tym później. Upowszechnienie zautomatyzowanej gospodarki materiałowej wyeksponowało potrzebę rzeczowego i normatywnego rozliczania zużycia materiałów. Dorobiono więc odpowiedni system użytkowy powiązany z „materiałówką".

Przedsiębiorstwo „ELBET" w Warszawie, odpowiedzialne za krajową, międzyresortową sprawozdawczość z produkcji wyrobów betonowych, dzięki rozwiązaniom informatycznym mogło się wreszcie uporać z należytym sporządzaniem materiałów dla GUS-u, a ponadto jest w stanie spożytkować zbierane informacje w przypisanej mu działalności koordynacyjnej. Są też już tacy dyrektorzy przedsiębiorstw, którzy bez etobowskich tabulogramów nie pojadą na żadną poważniejszą naradę.

Bywają jednak i sytuacje wręcz frustracyjne dla zespołu. Dotyczą zwłaszcza sfery planowania produkcji. Można próbować informatyzacji planowania operatywnego, ale przy tradycyjnej technice wsadowej z dowożeniem dokumentów do ośrodka obliczeniowego, trudno zejść poniżej dekadowego cyklu przetwarzania. Jeśli takie planowanie traktować szczegółowo, to nazbyt często staje się bezwartościowe, najczęściej z powodu zakłóceń w dostawach materiałów. Jeśli bardziej ogólnie — traci sens założona operatywność. Bez urządzeń końcowych w przedsiębiorstwach jest to niemal kwadratura koła i trudno się dziwić, że po paru próbach przedsiębiorstwa produkcyjne z takich usług po prostu rezygnują.

Zespół twierdzi, iż w tej dziedzinie zastosowań informatyki trudno o jakąś satysfakcję zawodową. I nie jest to ani przykład, ani pogląd odosobniony. Nie tylko w budownictwie, jak sądzę.

Twierdzą też dziewczyny — a doc. Cz. Uhma podobne ma zdanie — że osiągnięte dokonania są wynikiem nieomal godzenia ognia i wody, czyli improwizacji i konsekwencji w działaniu. Niedawno dyrekcja Zjednoczenia Przemysłu Betonów całkiem formalnie zarządziła, że opracowane systemy informatyczne mają być obowiązkowo stosowane w przedsiębiorstwach. W centrali zjednoczenia utworzono stanowisko specjalisty ds. zastosowań informatyki (też zresztą kobieta). Może to więc przełom w sprawie?

Pora zatem, aby dokładniej przedstawić zbiorowy tym razem obiekt niniejszego portretu. Jest to działający w warszawskim przedsiębiorstwie ETOB Zespół Projektowania Systemów dla Przemysłu Betonów, złożony z dwunastu kobiet. Można w nim umownie wyróżnić personalną bazę i nadbudowę.

„Bazę" stanowi siedem pań, pracujących w zespole od początku jego istnienia. Dwie z nich — kierująca zespołem Mieczysława Klochowicz oraz Wacława Michałowska — wywodzą się z politechnicznej specjalności organizacji i zarządzania w przemyśle, dwie — Krystyna Dobrowolska i Irena Piechota — w przydatnych dla informatyki specjalnościach SGPiS ukończyły, a dwie — Halina Brzezińska i Agnieszka Anacka — są absolwentkami numeryczno-komputerowych specjalności Uniwersytetu Warszawskiego. Wspiera je Julita Gozdan, łącząc znajomość elektroniki z programowaniem komputerów.

W skład „nadbudowy" wchodzą osoby, które później dołączyły do zespołu: Barbara Popławska — zaczęła pracę po liceum ogólnokształcącym, obecnie już po SGPiS, Jolanta Nagórska, Anna Zagalska i Izabela Gruszecka (SGPiS stacjonarne) oraz Ewa Wiatrowska — specjalizująca się w programowaniu.

— Co Panie chciałyby znaleźć w rekapitulacji tego portretu?

— *Dwie zwłaszcza rzeczy* — pada chóralna niemal odpowiedź — *to, że jesteśmy do poświęceń nawet gotowe, gdy na zadawane nam prace jest autentyczne zapotrzebowanie, i to, abyśmy miały we własnym przedsiębiorstwie bardziej operatywny dostęp do komputera dla prac programistycznych i wdrożeniowych.*

A na propozycje uzupełniania składu zespołu zawsze swego dyrektora pytają: — *czy aby nie kolejna kobieta?*

(wład)

(Hiper)wtajemniczenie teoriografowe

Pogląd głoszony przez B. Korzana [1]), że „z matematycznego punktu widzenia teoria grafów nie wnosi nic nowego", dziwi, a co najmniej zastanawia każdego, kto kiedykolwiek interesował się grafami. Ponieważ książka Korzana jest do pewnego stopnia uzasadnieniem takiego sądu jej Autora, nic więc dziwnego, że A. B. Empacher [2]) umieścił to zdanie jako motto swojej recenzji, w której stara się jeszcze bardziej utwierdzić Czytelnika w przekonaniu, że tak jest w rzeczywistości. (Przyjmuję, że Czytelnik potrafi łatwo odróżnić kiedy mowa jest o grafach, ich teorii i zastosowaniach, a kiedy grafy są jedynie ilustracją — najczęściej graficzną — rozwiązań).

Na upartego, to, o czym się mówi w teorii grafów, można podzielić na rozłączne części, sformułować w innych terminach i rozparcelować pomiędzy takie działy matematyki, jak kombinatoryka, topologia, geometria, teoria liczb i optymalizacja dyskretna, a wtedy sama teoria grafów przestanie głosić cokolwiek oryginalnego. Zgodzić się jednak ze mną muszą i Autor książki i Recenzent, że to, co wnoszą te części teorii grafów do innych działów matematyki, a co zostało wprowadzone i rozwinięte w języku grafów, jest wkładem właśnie tej teorii do matematyki. O tym, że tak jest, można się przekonać nie sięgając wcale do „matematycznych głębi" (ABE) teorii grafów, która, jak przypuszczam, musiała przerazić obu moich Respondentów. Dawne to już czasy, gdy teorię grafów uznawano za slumsy topologii, a o Königu mawiano *He is great in his art but his art is so small.* Obecnie, w dobie szybkiego rozwoju metod matematyki dyskretnej, teoria grafów ma już swoje ugruntowane miejsce.

Książka Korzana nie jest niestety ani lekturą uzupełniającą, ani z grupy tych, o których można powiedzieć, że wypełniają lukę (nie tylko tytułem) wśród dotychczasowych publikacji. Czy szanujący się informatyk umieści ją zatem w swej oszklonej biblioteczce, jak zaleca Recenzent, jest bardziej kwestią gustu niż oceną jej wartości. Szczegółowe uwagi o tej książce ukażą się w „Wiadomościach Matematycznych", a tutaj chciałbym zamieścić jedynie kilka słów, do napisania których skłoniły mnie „wtajemniczenia teorio(hiper)grafowe" Recenzenta.

Okazuje się, że tak jak wieloletnie wykładanie przedmiotu niekoniecznie predysponuje do pisania podręcznika, tak i monopol na niemal comiesięczne recenzje nie jest gwarancją ich poziomu. Nie wiem komu i jakie korzyści przynoszą takie opinie. Przypuszczam, że mimo przychylności Recenzenta, Jego uwagi rozczarowały nawet samego Autora omawianej książki.

Popatrzmy na szczegóły, w kolejności, w jakiej pojawiają się w recenzji.

Książka Korzana nie jest napisana niestety pod kątem żadnego odbiorcy. W sensie zastosowań jest nijaka, ani dla ekonomisty, ani dla genetyka, ani dla lingwisty, ani dla elektryka, i ani dla informatyka. Autor jedynie wspomina o pewnych problemach i algorytmach, które mogą mieć zastosowanie w innych dziedzinach. Żaden aspekt teorii grafów i jej zastosowań nie został w tej książce specjalnie wyróżniony. Wśród wymienionych przez Recenzenta dziedzin zastosowań, rzekomo uwzględnionych przez Korzana, są takie, które nie zostały skwitowane nawet przykładem! Dla zapoznania się z elementarnymi zastosowaniami teorii grafów polecam więc np. dla ekonomisty — **G. Avondo-Bodino, Economic Applications of the Theory of Graphs, London-Glasgow 1962 (tłum. ros. 1966);** dla genetyka — **F. S. Roberts, Discrete Mathematical Models, Prentice-Hall, 1976;** dla lingwisty — **L. Nebeksy, Graph Theory and Linguistics,** w: **L. W. Beineke, R. J. Wilson, Applications of Graph Theory, Academic Press 1979;** dla elektronika — **S. Seshu, M. B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, Reading, Mass. 1961 (tłum. ros. 1971) i W. K. Chen, Applied Graph Theory, North-Holland, Amsterdam 1971;** i dla informatyka — **D. E. Knuth, The Art of Computer Programming, Addison-Wesley, Reading, Mass., Vol. 1, 1968 (tłum ros. 1976)** oraz **A. V. Aho, J. E. Hopcroft, J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass. 1974 (tłum ros. 1979).** Przekonanie Recenzenta, że dotychczasowe zastosowania teorii grafów sprowadzają się jedynie do posługiwania się pojęciami tej teorii a nie jej wynikami, bierze się, przypuszczam, z powierzchownej znajomości zarówno samej teorii, jak i jej zastosowań.

Z pierwszych słów Recenzenta trudno jest wywnioskować, czy problem czterech kolorów jest matematycznym szczytem teorii grafów, gdyż tak długo opierał się wysiłkom pokoleń (chociaż nadal nie ma całkowitej pewności czy dowód Appela i Hakena jest poprawny), czy dlatego, że przyczynił się do rozwoju samej teorii i jej metod. Dalszy fragment recenzji rozwiewa jednak te wątpliwości, czytamy bowiem „rozwiązanie problemu nie wniosło specjalnie nic nowego". W tym miejscu zamiast komentarza i pojedynczych przykładów polecam całą ich moc w książkach **O. Ore, The Four Color Problem, Academic Press, N.Y. 1967** i **T. Saaty, P. Kainen, The Four Color Problem: Assults and Conquest, Mc Graw-Hill, 1977.**

Na czym ma polegać „matematyczna finezja" (ABE) problemu czterech barw? Recenzentowi wydaje się, że skoro pięć kolorów zawsze wystarcza (jest to twierdzenie Heawooda, a ostatnio D. Matula pokazał jak można efektywnie malować każdą mapę co najwyższej pięcioma kolorami), to nie ma potrzeby walczyć tak zażarcie o ten jeden kolor mniej. Idąc tym samym tropem, za matematyczną finezję można także uznać problem kolorowania połączeń grafu (twierdzenie Vizinga jest bowiem oszacowaniem z dokładnością do jednego minimalnej liczby kolorów potrzebnych w tym przypadku) oraz nieskończoną liczbę innych problemów kombinatorycznych polegających na wyznaczeniu największej lub najmniejszej liczby elementów pewnych zbiorów skończonych. Stąd już blisko do degradacji optymalizacji dyskretnej, w której aż ciasno od długowiecznych problemów nie rozwiązanych. Przypuszczam, że degradacja matematyki w tym przypadku została spowodowana siłą narzędzi (tutaj maszyn cyfrowych), jakimi dysponują dzisiaj informatycy, chociaż truizmem jest stwierdzenie, że potęga maszyn cyfrowych tkwi nie w ich wielkości fizycznej, a w sposobach ich wykorzystywania i stosowanych metodach.

Żadne monograficzne opracowanie **O. Ore** nie zostało niestety przetłumaczone na język polski. Przypuszczam, że Recenzent nie miał w rękach żadnej książki Ore, ani tej wydanej po polsku **(Wstęp do teorii grafów, PWN Warszawa 1966),** ani innych opracowań tego autora. Ta pierwsza to po prostu bardzo elementarny wstęp do teorii grafów na poziomie naszych opracowań dla szkół podstawowych i średnich, poziom monograficzny natomiast reprezentują książki dotychczas nie przetłumaczone **Theory of Graphs, Providence, XXXVIII, 1962** i wspomniana już **The Four Color Problem, Academic Press, N.Y. 1967.**

Przekonanie Recenzenta o „świeżości monograficznej" książki Korzana płynie niestety z nieznajomości obecnego (lub nawet ówczesnego, tj. z 1976 r.) stanu teorii grafów, jej algorytmów i zastosowań.

Chyba najrzetelniej potraktowany został w recenzji Dodatek I z omawianej książki poświęcony algebrom Boole'a, w którym Recenzentowi nie podoba się, że rozważania zostały ograniczone do algebry dwuelementowej, a nie ma mowy o algebrach skończenie-wieloelementowych i twierdzeniu, że liczba tych elementów musi być zawsze całkowitą potęgą liczby 2. Tylko co ta finezja matematyczna ma wspólnego z grafami i ich zastosowaniami, jeśli nawet elementarne fakty i metody algebr Boole'a są mało praktyczne i nie znalazły zbyt wielkiego uznania ani u teoretyków, ani u praktyków teorii grafów.

Tymi uwagami nie bronię książki Korzana, bo trudno jest obronić ją przed wieloma zarzutami, ani samych grafów, których bronić nie trzeba. Chciałem jedynie wyrazić swój sprzeciw przeciwko takim recenzjom z pogranicza „wtajemniczeń" i nieznajomości przedmiotu recenzowanej pracy.

Maciej M. SYSŁO

[1]) Korzan B.: Elementy teorii grafów i sieci. Metody i zastosowania. WNT, Warszawa 1978

[2]) Empacher A. B.: Wtajemniczenia teorio(hiper)grafowe. INFORMATYKA, nr 8—9/1979

Komputeryzacja wielkich miast – WARSZAWA

Rozwój współczesnych miast, powodujący coraz więcej zakłóceń w tak wielu dziedzinach naszego życia, powinien być świadomie i umiejętnie sterowany. Znakomitym narzędziem takiej „miastotwórczej" polityki są systemy informatyczne, dzięki którym uzyskuje się obiektywny i — co najważniejsze — aktualny obraz miasta. Kompleksowe plany rozwoju aglomeracji, obsługa ludności, usprawnienie administracji i gospodarki miejskiej — wymagają coraz większej ilości informacji, coraz bardziej precyzyjnych i coraz szybciej przekazywanych. Metody tradycyjne stały się już nieskuteczne.

Urbaniści wyróżniają dwa podstawowe czynniki, determinujące prawidłowy rozwój miast: efektywny układ komunikacyjny oraz właściwe rozmieszczenie ludności. Opisanie obu tych elementów (zapis cyfrowy) jest sprawą zasadniczą dla miejskich służb informatycznych. Elementy te stały się kanwą odczytu mgr Jerzego Trybulskiego, dyrektora Stołecznego Ośrodka Elektronicznej Techniki Obliczeniowej (SOETO), odczytu zorganizowanego przez Komisję Informatyki ZG PTE: „Komputeryzacja w dużych aglomeracjach miejskich na przykładzie Warszawy". Zanalizowane zostały problemy stołeczne — na podstawie Programu Rozwoju Informatyki w Warszawie i Województwie w latach 1980—85.

Wykorzystywane w Warszawie „operacyjne" systemy informatyczne — opisujące wycinki trzech głównych zbiorów aglomeracji: ludność, zasoby techniczne miasta (budynki, sieci, środki transportu), procesy (zaopatrzeniowe, komunikacyjne, inwestycyjne, remontowe) — mogą wspólnie stworzyć bank informacji. Będzie to jednak możliwe dopiero wówczas, gdy zostanie opracowana w układzie cyfrowym sieć przestrzenna miasta (inaczej — urbanistyczna lub geodezyjna), która przyporządkowuje elementy aglomeracji punktom na jej planie. Dzięki takiej osnowie można dokładnie analizować wpływ zaistniałych procesów na globalny stan miasta, a w efekcie — określać cele i metody „strategiczne".

Te miasta na Zachodzie, w których układ przestrzenny i oparty na nim bank danych zostały już opracowane, bronią się skuteczniej przed chaotycznym i przypadkowym rozwojem. Decyzje inwestycyjne są tam podejmowane tylko w przypadku, gdy nie grożą zakłóceniem układu komunikacyjnego bądź pogorszeniem życia mieszkańców.

W kilku też krajach wprowadzono systemy ewidencji ludności (USA, Australia, Szwecja, Bawaria w RFN). Adres każdej osoby jest tu powiązany z odpowiednim punktem na planie miasta. Dzięki temu uzyskuje się na przykład możliwość wyznaczania optymalnych dróg dojazdowych do miejsca pracy i nauki czy też optymalnej sieci usług.

W Polsce, mimo konsekwentnych prac w tym kierunku, nie udało się — jak dotąd — stworzyć żadnego systemu zbiorczego dla większych aglomeracji. Jednym z głównych powodów jest zapewne niedobór sprzętu. Wymowne jest

w tym wypadku porównanie ilości komputerów klasy R-32 (umownych), używanych w procesie zarządzania gospodarką miejską (na 100 tys. mieszkańców): Marsylia — 4,0, Zurich — 1,5, Warszawa — 0,13.

Warszawska, miejska informatyka ma już za sobą 13-letni okres istnienia (1967 — powstanie SOETO), na koncie zaś — kilka udanych systemów „operacyjnych". Niestety wszystkie dotychczasowe prace mają charakter przyczynkarski i niesystematyczny.

Spośród systemów „operacyjnych", zaspokajających bieżące potrzeby miasta, warto wyróżnić cztery:

- system SARCZ do obliczania i egzekwowania czynszów w budynkach, pozostających w gestii miasta (nie spółdzielczych) — system największy, pochłaniający moc obliczeniową równą ponad połowie mocy systemu ODRA 1305 z pamięcią 128 K słów
- system gospodarki materiałowej, stosowany we wszystkich jednostkach gospodarczych podległych Urzędowi Miasta (przedsiębiorstwa miejskie i rejony obsługi mieszkańców)
- system MANY, rozliczający mandaty kredytowe
- systemy dla MZK, obejmujące gospodarkę materiałową, gospodarkę remontową taboru, gospodarkę częściami zamiennymi, obliczanie płac personelu oraz ewidencję kadr.

Funkcjonują też trzy systemy o szerszym zasięgu, stale modernizowane, które w przyszłości mogą ułatwić działania „strategiczne".

- System EBIR (Elektroniczny Bank Inżynierii Ruchu). Jego podstawą jest sieć ulic (kolejno ponumerowane skrzyżowania; odcinki dróg określone numerami skrzyżowań, które są przez te odcinki łączone), w którą wpisano dane o urządzeniach i znakach drogowych, sygnałach, nawierzchni jezdni, itd. Sieć aktualizowana jest metodą wsadową raz w miesiącu. Na potrzeby systemu (ale poza nim) prowadzi się analizę ruchu drogowego (okresy natężeń, miejsca newralgiczne) oraz rozmieszczenia pojazdów (parkingi, postoje, miejsca zagęszczeń). Programy symulujące zamieniają następnie wskazania dyskretne — w ciągłe, te zaś wykorzystywane są do obliczeń. Aktualnie EBIR umożliwia wybór miejsc wymagających najszybszego remontu, wyznaczanie optymalnych tras dla pojazdów ponadgabarytowych oraz optymalnych objazdów, zapis miejsc i przyczyn wypadków drogowych.
- Grupa programów do prognozowania i symulacji, które są użytkowane przez Biuro Planowania Rozwoju Warszawy.
- System ewidencji lokali biurowych w dzielnicy Śródmieście, wykorzystujący macierz instytucji i ich adresów. Niebawem system ten zostanie rozszerzony na wszystkie dzielnice.

W roku 1975, korzystając z zagranicznych doświadczeń, rozpoczęto pracę nad koncepcją Banku Informacji o Warszawie, umożliwiającego stworzenie zintegrowanego systemu informowania stołecznych władz. Cel ten jednak nie zostanie osiągnięty, jeśli nie uda się uporządkować systemów „operacyjnych" niższych szczebli. Brak cyfrowego zapisu sieci przestrzennej miasta, o której wspomniany Program wspomina bardzo enigmatycznie, oraz dezintegracja drobnych systemów (nawet w obrębie jednego zjednoczenia) uniemożliwiają agregowanie informacji, a dalej — zarządzanie w skali makro.

W ciągu najbliższych pięciu lat ma nastąpić — jak się zakłada — istotna unifikacja systemów przyporządkowanych poszczególnym obiektom, tak by ujednolicone informacje mogły zasilić wspólną bazę danych. Następnie — w zależności od dostępnych środków i od konsekwencji w działaniu — powstanie bank danych oraz system informowania kierownictwa Urzędu Miasta. Warunkiem koniecznym tego systemu jest umożliwienie pracy konwersacyjnej z maszyną, a także znaczne zwiększenie liczby jednostek pamięci dyskowych, a więc rozbudowa posiadanego sprzętu.

Wszystkie działania, wprowadzające informatykę do procesów zarządzania w administracji i gospodarce Miasta, mają być w najbliższym czasie podporządkowane koncepcji Banku Informacji o Warszawie. Główne systemy BIW, oczekujące na realizację, to:

- system ewidencji ludności — PESEL, stanowiący podstawę systemów o charakterze gospodarczym (analiza rynku

pracy, pośrednictwo pracy, bilanse siły roboczej, analizy i prognozy demograficzne, itp.)
- system ewidencji gruntów, stanowiący identyfikator przestrzenny dla systemów ewidencyjnych z zakresu infrastruktury technicznej
- system ewidencji budynków
- system ewidencji stanu technicznego lokali i sposobu ich użytkowania
- systemy ewidencji infrastruktury technicznej, a ściśle — sieci wodno-kanalizacyjnej, ciepłowniczej, gazowej, itp.

Systemy powyższe mogą w istotny sposób wpłynąć na ostateczny kształt kompleksowej komputeryzacji Warszawy. Aby jednak na ich bazie mógł powstać przewidywany Bank Informacji, koniecznym staje się ustalenie (przez Ministerstwo Administracji, Gospodarki Terenowej i Ochrony Środowiska) zasad cyfrowego opisu układu przestrzennego miasta.

W perspektywicznym programie zaproponowano następujący kształt organizacyjny stołecznej informatyki:
- potencjał komputerowy, oparty na maszynach RIAD, przystosowanych do pracy w systemach teleprzetwarzania, koncentrowany będzie w SOETO
- w przypadku instalowania maszyn przez instytucje miejskie poza SOETO, użytkownik przekaże do SOETO limity inwestycyjne, które zostaną przeznaczone na zorganizowanie i obsługę ośrodka na jego terenie
- użytkownicy będą wyposażeni (od 1980 r.) w zdalne stacje abonenckie, zapewniające dostęp do komputerów w SOETO
- stacje przygotowania danych będą decentralizowane i obejmowane serwisem technicznym SOETO
- SOETO będzie typizować systemy, programy i standardy eksploatacyjne dla wszystkich jednostek podległych Prezydentowi Miasta
- kadra projektantów i programistów systemowych będzie skoncentrowana w SOETO
- projektowanie struktur funkcjonalnych systemów będzie się odbywać poza SOETO — u poszczególnych użytkowników.

Program obejmujący najbliższe pięciolecie nie jest programem maksimum, umożliwia jedynie zaspokojenie najpilniejszych, doraźnych potrzeb. Ma też na celu stworzenie bazy, która dawałaby szansę na zaspokojenie wszystkich potrzeb w latach następnych. Na ile realne są przewidywania zawarte w Programie — przyszłość pokaże. Jego realizacja wymaga w każdym razie nakładów inwestycyjnych w wysokości ok. 200 mln zł oraz ok. 1100 m² powierzchni na instalację sprzętu.

Systemy informatyczne nie zmienią oczywiście w sposób istotny trudnej sytuacji miasta (np. w komunikacji — brak metra, 150 tys. samochodów bez miejsca na postój, niedobory w taborze MZK, przewężenia ulic w Śródmieściu...). Są natomiast w stanie zoptymalizować wykorzystanie możliwości dzisiejszych, a na przyszłość — wskazać rozwiązania dogodniejsze, bardziej radykalne. Warszawa, idąc śladem wielkich metropolii, pragnie przecież wykorzystać ich doświadczenia.

Opracował Z.G.

„Matematycy bliżej przemysłu, przemysł bliżej matematyki"

— pod takim hasłem organizowane są coroczne ogólnopolskie konferencje zastosowań matematyki. Taki jest również sens emblematu ostatniego sabatu matematyków *), przedstawiającego „fabrykę wypełnioną matematyką". Z komina tej fabryki ulatuje wolny od skażeń biały obłok nieszkodliwego dymu, a nie jak niegdyś, z komina „tradycyjnej" fabryki — czarny trujący dym. Nie należy oczywiście przesadzać: sami matematycy nie rozwiążą jednak wszystkich potrzeb ochrony środowiska naturalnego.

W tym miejscu wypada zrobić dygresję o przeszło wiekowych już próbach podziału matematyki na „czystą" i „stosowaną". Sztuczność wszelkich takich podziałów wyraża się przede wszystkim w tym, że jeden i ten sam matematyk w pewnych momentach swej działalności twórczo-naukowej jest „czystym" teoretykiem, w innych zaś próbuje znaleźć zastosowanie dla swych własnych czy też innych teoretycznych wyników.

Należy jednak pamiętać, że zastosowania matematyki w przemyśle — czy też ogólniej mówiąc, w gospodarce narodowej — mogą się nie-matematykom wydać trudne i przeteoretyzowane. Jest to może pewnym mankamentem dotychczasowych konferencji, że nie starały się szerzej popularyzować matematyki, poprzestając na formie seminaryjnej zajęć. W tej bowiem sytuacji trzeba pewnej odwagi, aby nie będąc zawodowym matematykiem zasiąść wśród wtajemniczonych. Tym bardziej warto podkreślić udział zawodowego lekarza (w dodatku płci pięknej) w ostatniej konferencji — jako współautora referatu o ustaleniu przynależności filogenetycznej kręgowców kopalnych.

Zaletą dotychczasowych konferencji jest wysoki poziom naukowy wygłaszanych referatów i komunikatów (wydawanych drukiem w formie streszczeń i liczących się w dorobku naukowym autorów). Konferencje zastosowań matematyki spełniają więc ważną rolę stymulującą, zachęcając, zwłaszcza młodszych matematyków, do spojrzenia na problematykę macierzystego zakładu pracy z naukowo-matematycznego punktu widzenia. W ten sposób powstały np. referaty: „Minimalizacja kosztów transportu wybranego asortymentu prefabrykatów z fabryki domów na budowy", „Cykliczna odnowa stałego zapasu towarów ulegających degradacji", „Algorytm kolejności walcowania pasm w systemie automatycznego zarządzania walcownią blach grubych", „Model matematyczny rocznego asortymentowego planu produkcji zjednoczenia BUMAR". Trzeba przyznać, że nie są to proste, zrozumiałe tytuły.

Przykładem bardziej „czystych" wyników naukowych, referowanych na konferencji, może być nowe kryterium niezależności zmiennych losowych, szczególnie dogodne w rachunkach numerycznych. Referowano również algorytmy rozwiązywania równań różniczkowych opisujących mokre formowanie włókien akrylonitrylowych i acetylocelulozowych, zastosowane w programie napisanym na minikomputer HP-9820. Oczywiście, nie sposób tutaj wymienić wszystkich tematów poruszonych na konferencji, gdyż same streszczenia referatów liczą blisko 90 stron druku.

Istotnym uzupełnieniem konferencji jest tradycyjna dwudniowa sesja problemowa organizowana pod hasłem: Kto wie, jak to rozwiązać? Szereg prac referowanych na późniejszych zjazdach naukowych — w szczególności na konferencjach zastosowań — wzięło swój początek właśnie z takich „nieuczesanych dyskusji" nad pomysłami rozwiązań. Trzeba bowiem pamiętać, że podstawą sukcesów w praktycznych zastosowaniach matematyki jest przede wszystkim próba ścisłego sformułowania problemu, a następnie dobrania doń odpowiedniego modelu obliczeniowego.

Drugim ważnym uzupełnieniem tej „nie-czystej" imprezy matematycznej są spotkania uczestników konferencji z lokalnymi środowiskami przemysłowymi oraz szkolnymi. To jest właśnie ów pożądany nurt popularyzacyjny — bez obniżania poziomu naukowego samej konferencji. Z tego właśnie względu organizatorzy lokalizują ogólnopolskie konferencje zastosowań matematyki zawsze w innym województwie. W ten sposób nawet małe szkoły i zakłady mają szansę na bezpośredni kontakt z najwybitniejszymi naukowcami. Czas najwyższy by zrozumieć i docenić rolę matematyki we współczesnym świecie.

Ostatnią konferencję trafnie umiejscowiono w olbrzymim domu wypoczynkowym Banku Gospodarki Żywnościowej (nomen-omen, wyżywienie było „na medal"). Toteż około 200 matematyków krajowych i kilku gości zagranicznych (z Drezna, Sofii i Wilna) znalazło tu rzeczywiście wspaniałe warunki do wzajemnych kontaktów naukowych.

Kolejna, IX konferencja będzie zorganizowana na terenie łódzkim. Można mieć nadzieję, że bliskość wielkiego miasta i dogodne połączenia kolejowe ze stolicą przyciągną większą liczbę gości zagranicznych. Można też spodziewać się większego udziału „branżystów" zarażonych bakcylami matematyki i informatyki, przecierających ścieżki dla nowych zastosowań.

(ABE)

*) Ósma Ogólnopolska Konferencja Zastosowań Matematyki odbyła się w Dąbkach k.Darłowa, w dniach 8—17 października 1979 r., pod patronatem Komitetu Nauk Matematycznych Polskiej Akademii Nauk, Zarządu Głównego Polskiego Towarzystwa Matematycznego oraz Instytutu Matematycznego PAN

W poszukiwaniu rozwiązań optymalnych

Na rynku księgarskim ukazała się kolejna publikacja o charakterze podręcznikowym, która porusza podstawowe problemy związane z budową, działaniem i oceną systemów komputerowych. Jest to tłumaczenie książki amerykańskiego autora, Borisa Beizera, pod polskim tytułem „Organizacja systemów komputerowych" *).

Od razu warto zastrzec, że jest to tłumaczenie tylko pierwszego tomu dwutomowej pracy Beizera, ale obie części książki są na tyle niezależne, iż mogą stanowić samodzielne pozycje. Tom pierwszy pomyślano jako podręcznik dla słuchaczy jednosemestralnego kursu informatyki, którzy mieli już z nią kontakt. Szczególny nacisk położono na problemy projektowania systemów komputerowych, przy czym tematem wiodącym jest poszukiwanie rozwiązań optymalnych poprzez ustanawianie kompromisów pomiędzy istniejącymi ograniczeniami oraz wyzyskiwanie pewnej zamienności niektórych składników (np. oprogramowania i sprzętu) i charakterystyk systemów cyfrowych (np. wielkości pamięci i czasu wykonania). Tom drugi (to informacja dla czytelników, którzy po lekturze polskiego tłumaczenia zechcą sięgnąć do oryginału) stanowi konsekwentną kontynuację tematu, a w szczególności zawiera praktyczne rozważania o niezawodności złożonych systemów cyfrowych i projektowaniu takich systemów, a w tym o metodach autodetekcji usterek i samonaprawialności. Jest tam też dużo wiadomości o realizacji systemów komputerowych.

Można sądzić, że jedną z pierwszych czynności czytelnika, który weźmie do ręki tłumaczenie książki Beizera będzie ustalenie daty wydania oryginału. Znajdzie on, że był to rok 1971, a więc jak na tempo rozwoju informatyki dość dawno. Jeśli nie zniechęci go to jednak do przejrzenia książki, przekona się, że pomieszczono w niej szereg interesujących wiadomości, co więcej — uczyniono to w sposób umiejętny i oryginalny. Oto krótki przegląd poszczególnych rozdziałów.

W rozdziale pierwszym podano opisową definicję systemu komputerowego, kilka przykładów takich systemów (SAGE, RW-400, CDC 7600, B-8500, ILLIAC IV itd.) oraz uwagi o architekturze i projektowaniu systemów komputerowych. Zwraca uwagę określenie systemu komputerowego, gdzie obok sprzętu i oprogramowania autor umieścił personel związany z eksploatacją systemu.

Rozdział drugi przedstawia najważniejsze składniki maszyn cyfrowych: sprzęt i oprogramowanie, a następnie zarysowuje fazy rozwoju celów, które stawiali sobie projektanci systemów cyfrowych. Rozdział ten jest szczególnie ważny, gdyż stanowi rodzaj przewodnika po dalszych partiach książki. Bardzo interesujące jest pojęcie cechy modalnej zbioru instrukcji, przez co autor rozumie własność, która w wyraźny sposób odróżnia instrukcje tego zbioru od pozostałych instrukcji z listy rozkazów. Wprowadzenie cech modalnych okazało się dobrym sposobem klasyfikacji instrukcji, a jednocześnie pozwala ujawnić, jak często producenci komputerów stosują „sztucznie" rozbudowane listy rozkazów, głównie w celach reklamowych.

W tym samym rozdziale Beizer wyłożył swoje podejście do architektury systemów. Uważa on, że najstosowniejsza jest „filozofia złotego środka", co w praktyce oznacza m.in. potrzebę określania wartości krańcowych różnorakich parametrów mających wpływ na rozwiązanie i poszukiwanie *rozwiązania optymalnego pomiędzy... rozwiązaniami skrajnie przeciwstawnymi* (s. 83). Wśród parametrów wymienia się: koszty, czas, dostępność sprzętu i oprogramowania, elastyczność, żywotność i starzenie się systemu. W konkluzji rozdziału drugiego można przeczytać: *System komputerowy różni się od zgrupowania sprzętu i oprogramowania tym, że stanowi optymalne połączenie tych dwóch i innych elementów* (s. 98). Wydaje się, że opinia ta może być dobrym pretekstem do dyskusji.

W rozdziale trzecim dużo miejsca poświęcono sprawom związanym z budową i działaniem jednostki centralnej oraz klasyfikacji komputerów na podstawie liczby adresów wymienionych w częściach adresowych większości instrukcji z listy rozkazów (komputery zero-, jedno-, dwu-, trzy- i czteroadresowe). Ponadto przedstawiono różne cechy modalne i odpowiadające im typy instrukcji, wprowadzono pojęcia instrukcji prostej i złożonej (jako kombinacji instrukcji prostych) oraz wskazano na listę rozkazów, stopień równoległości struktury sprzętu, liczbę rejestrów i mikroprogramowanie jako czynniki charakteryzujące moc obliczeniową komputera.

W kolejnym rozdziale omówiono różne rodzaje pamięci, z uwzględnieniem ich funkcji w systemie komputerowym oraz występującymi pomiędzy nimi relacjami kosztów. Zwrócono uwagę na to, że pamięci tworzą hierarchię: od pamięci o małej pojemności, dużej szybkości i małym opóźnieniu do pamięci o wielkiej pojemności, małej szybkości i dużym opóźnieniu. Omówiono koncepcje nakładania i przeplatania, stosowane w celu zwiększenia efektywności dostępu do pamięci. Następnie wprowadzono określenie przełącznika pamięci, który stanowi główny węzeł komunikacyjny wewnątrz systemu, po czym przedstawiono ideę przerwań. Te ostatnie dwa pojęcia zostały zaprezentowane jako środki zmniejszające straty czasów pamięci i jednostki centralnej. W drugiej części tego rozdziału podano określenie kanału oraz dokonano przeglądu urządzeń peryferyjnych oraz sprzętu telekomunikacyjnego używanego w systemach komputerowych.

O rozdziale piątym sam autor wyraził się, że *celem tego rozdziału jest wstępne zapoznanie czytelnika z folklorem programowania i wskazanie obowiązujących w nim zasad* (s. 278). Programowanie przedstawiono jako proces złożony z pięciu etapów: analizy funkcjonalnej, analizy programowej, kodowania, testowania i dokumentowania. Szczególnie wartościowe jest pomieszczenie uwag o dokumentowaniu programów, które bywa często niedoceniane i wręcz pomijane. Beizer jasno opowiada się za prowadzeniem starannej dokumentacji i wyraża to dosadnie: *Ci, którzy nie dokumentują swoich programów, nie są zawodowymi programistami* (s. 350). Ciekawym fragmentem tego rozdziału jest część poświęcona stylowi programowania. Za generalną zasadę w tym względzie podaje się konsekwentne przestrzeganie przyjętych przez programistę reguł. W rozdziale tym omówiono takie narzędzia analizy funkcjonalnej i programowej, jak tablice decyzyjne i przełącznice sekwencyjne. Zwrócono uwagę na podstawowy problem programowania: zamienność czasu realizacji programu i pojemności użytkowanej pamięci.

Rozdział szósty jest poświęcony oprogramowaniu firmowemu, które zdaniem autora *powstało jako historyczna konieczność w odpowiedzi na niemożliwy do zaspokojenia brak programistów* (s. 448). Określenie „historyczna konieczność" przywykliśmy używać w nieco innym kontekście, ale faktem jest, że stosowanie oprogramowania firmowego zmniejsza koszty ponoszone przez użytkownika, choć zarazem ogranicza jego swobodę. Do oprogramowania firmowego zaliczono: programy ładujące, assemblery, kompilatory, interpretery, programy diagnostyczne i analityczne, system operacyjny i bibliotekę programów standardowych używanych jako fragmenty innych programów lub do manipulacji programami i danymi.

Ostatni i zarazem najobszerniejszy rozdział nosi tytuł „Analiza". Objętość i zakres tego rozdziału nie dziwią, gdyż traktuje on o metodach optymalnego projektowania systemów komputerowych, a to przecież jest nadrzędnym celem książki. Metody te umożliwiają analizę statystyczną wykorzystania czasu pracy komputera, pojemności pamięci i kanałów oraz przeprowadzenie oceny programów. Dla oceny czasu wykonania i innych parametrów programu autor zaleca macierzową metodę eliminacji węzłów. Podział nakładów na różne fragmenty programów proponuje się oprzeć na metodach programowania liniowego, a wyznaczanie całkowitego czasu wykonania programu — na teorii masowej obsługi. Autor podaje pięć zasad analizy. Warto w tym miejscu zacytować choćby dwie ze względu na ich uniwersalny charakter: *Wiele drobnych przypuszczeń jest lepsze od jednego dużego* oraz *Dobra analiza często kończy się wcześniej niż się przypuszcza, a niekiedy w ogóle się nie odbywa* (s. 581). Jako żywo przypomina to znane skądinąd zasady Petera. Można by, jako szóstą, ustanowić żartobliwą zasadę: Jeśli nie wiesz, co analizujesz, to zapewne zanalizujesz coś innego (parafrazując jedno ze zdań z „Alicji w krainie czarów").

*) **Boris Beizer: Organizacja systemów komputerowych. PWN, „Biblioteka Informatyki", Warszawa 1979. Z języka angielskiego tłumaczyli: Mieczysław Muraszkiewicz, Zbigniew Nowicki, Marek Urbański. (Dane oryginału: Boris Beizer: The Architecter and Engineering of Digital Computer Complexes, vol. 1. 1971 Plenum Press, New York). Wydanie I, nakład 6000 + 280 egz., ark. wyd. 29,0, stron 588, cena: 60 zł**

Wszystkie rozdziały opatrzone są ciekawymi zadaniami i problemami do samodzielnego rozwiązania oraz strannie dobraną i skomentowaną literaturą. Z uwagi na rok wydania oryginału wiele pozycji umieszczonych przez Beizera ma już tylko znaczenie historyczne.

Polski czytelnik książki Beizera zapewne zada pytanie o jej miejsce wśród innych podobnych pozycji dostępnych na naszym rynku. Odpowiedź warto poprzedzić opinią R. W. Hamminga — autora słowa wstępnego omawianej książki: *Właściwe podejście do informatyki leży oczywiście pomiędzy tymi dwoma skrajnymi podejściami* (matematycznym i inżynierskim — przyp. M. M) *i polega na połączeniu w sposób uzupełniający się technik matematycznych oraz rozsądku inżyniera* (s. 13). Beizerowi udało się zachować równowagę pomiędzy praktycznymi aspektami dyskutowanych tematów a ścisłością ujęcia, przy czym osiągnął to bez wprowadzenia rozwiniętego aparatu matematycznego. Udało się mu nawet coś więcej: zbliżył się do wytworzenia pewnej otwartej metodologii projektowania systemów komputerowych, która może być wzbogacana doświadczeniem i intuicją projektanta. Ponadto zachęca on do spojrzenia na problemy informatyczne przez pryzmat innych dziedzin. Przykładowo, rozważając zagadnienia projektowania systemów komputerowych cytuje i rekomenduje prace W. Gropiusa — założyciela architektonicznej szkoły „Bauhaus". W innym miejscu powołuje się na C. N. Parkinsona i odsyła do jego praw, często sięga nawet do anegdoty. W sumie jest to książka, która tematycznie w dużym stopniu jest zbieżna z wydanymi w kraju pracami, natomiast w sposobie ujęcia problemów i żywej formie narracji nie ma wielu pozycji konkurencyjnych.

Książka Beizera ma swoisty klimat, który — jak można sądzić — jest wynikiem jego pragmatyzmu zmieszanego ze skłonnością do uogólnień i poszukiwania metody. Autor posłużył się różnymi środkami, by dotrzeć do czytelnika, choćby wspomnianą już anegdotą. Wypada więc wyrazić żal, że w polskim tłumaczeniu nie znalazły miejsca bardzo pieczołowicie i celnie dobrane motta poprzedzające kolejne rozdziały.

Można przypuszczać, że „Organizacja systemów komputerowych" będzie cenną książką zarówno dla początkujących, jak i zaawansowanych informatyków. Tym pierwszym dostarczy usystematyzowanej i praktycznej wiedzy, tym drugim — przyjemności.

Mieczysław MURASZKIEWICZ

Międzynarodowe konferencje

Sztuczna inteligencja i robotyka

Rada Naukowa sekcji „Sztuczna inteligencja" Komitetu Analizy Systemowej Prezydium Akademii Nauk ZSRR oraz Instytut Cybernetyki Technicznej Słowackiej Akademii Nauk organizują w dniach od 30.6 do 4.7.1980 r. międzynarodową konferencję naukową pt. „Sztuczna inteligencja i systemy informacyjno-sterujące robotów". Konferencja, przewidziana dla ograniczonej liczby ok. 100 uczestników, odbędzie się w Domu Naukowca mieszczącym się w Smolencach k. Bratysławy.

Przewodniczącym Komitetu Programowego konferencji jest prof. G. S. Pospiełow z Centrum Obliczeniowego AN ZSRR, zaś członkami 18 wybitnych naukowców z krajów socjalistycznych. Polskę reprezentuje prof. Z. Pawlak z Instytutu Podstaw Informatyki PAN.

Celem konferencji jest zapoznanie uczestników z aktualnym stanem oraz najnowszymi osiągnięciami naukowymi w dziedzinie sztucznej inteligencji, systemów informacyjno-sterujących robotów oraz specjalizowanych komputerów dla potrzeb sztucznej inteligencji i robotyki. Program konferencji obejmuje zamówione referaty przeglądowe, referaty specjalistyczne zgłoszone przez uczestników oraz dyskusje panelowe, a także wystawy prezentujące dorobek naukowy uczestników. Materiały konferencyjne zostaną opublikowane przez Wydawnictwo Słowackiej Akademii Nauk. Językami oficjalnymi konferencji będą rosyjski i angielski.

Komitet programowy ustalił następujące główne tematy konferencji:

- problemy teoretyczne sztucznej inteligencji
- systemy dostrzegania i rozpoznawania otoczenia przez roboty
- systemy sterowania robotów
- komunikowanie się z komputerami w języku naturalnym
- specjalizowane systemy komputerowe oraz sprzęt dla potrzeb sztucznej inteligencji i robotyki.

W czasie trwania konferencji w Brnie zostanie zorganizowana przez Czechosłowackie Stowarzyszenie Nauki i Techniki międzynarodowa wystawa ROBOT 80 o podobnej tematyce.

Komputeryzacja w bankach

Francuski Instytut Badań Bankowości INSIG organizuje w dniach 23—25 kwietnia br. w paryskim hotelu HILTON-SUFFREN dziewiątą doroczną międzynarodową konferencję pt. „Banki i komputery". Tematem konferencji będzie organizacja pracy banków w latach dziewięćdziesiątych, ze szczególnym uwzględnieniem coraz głębiej postępującej komputeryzacji operacji bankowych.

Konferencji będzie towarzyszyła międzynarodowa wystawa specjalistycznego sprzętu komputerowego dla obsługi działalności banków, zlokalizowana w salach recepcyjnych tego samego hotelu.

Działalność IFIP

Rada i Ogólne Zgromadzenie IFIP zebrały się w Londynie na zaproszenie British Computer Society na specjalnym posiedzeniu bezpośrednio przed rozpoczęciem konferencji Euro — IFIP 79 *). Przedmiotem posiedzenia były wybór nowego prezydium, przyjęcie nowych krajów członkowskich, sprawy finansowe oraz polityka dalszego rozwoju Federacji.

Na przewodniczącego Federacji wybrano ponownie na okres następnej kadencji (1980—1983) prof. P. A. Bobilliera (Szwajcaria). Jego poprzednik dr R. I. Tanaka (USA) otrzymał tytuł członka honorowego IFIP, jako trzeci zasłużony działacz IFIP po I. L. Auerbachu i prof. H. Zemanku. Wiceprzewodniczącym został p. Jussi Tuori (Finlandia) po 6 latach piastowania funkcji skarbnika Federacji, na którym to stanowisku zastąpił go p. Knut Hernaes (Szwecja). Sekretarzami Federacji wybrano p. G. J. Morrisa (W. Brytania) oraz dr R. C. de Oliveira (Brazylia).

Liczba członków Federacji wzrosła do 39 dzięki przyjęciu z ważnością od stycznia 1980 r. następujących krajów:

- Chińskiej Republiki Ludowej, którą reprezentować będzie Chiński Instytut Elektroniki (CIE)
- Maroka, reprezentowanego przez Marokańskie Stowarzyszenie Rozwoju Elektroniki, Informatyki i Automatyki (AMADEIA).

Wg sprawozdania skarbnika sytuacja finansowa Federacji jest pomyślna, wykazując w pełni zrównoważony budżet z odpowiednimi rezerwami. Finanse Federacji uzależnione są głównie od świadczeń krajów członkowskich oraz nadwyżek z organizowania cyklicznych kongresów.

Ogólne Zgromadzenie IFIP powołało Komitet Rozwoju Perspektywicznego (Future Policy Committee) — pod przewodnictwem prof. R. Marasimhana (Indie) celem dokonania przeglądu dotychczasowego zakresu działalności i zdecydowania ewentualnych zmian lub rozszerzeń, gwarantujących nadążanie za szybkimi zmianami w informatyce. Szczególny nacisk postanowiono położyć na rolę Federacji w stosunku do krajów rozwijających się. **(W.K.)**

*) O konferencji tej donosiliśmy w numerze 12/79 INFORMATYKI

GRZEGORZ GRUCHMAN
ZETO Poznań

Problemy budowy wspólnych baz danych

Pozycja technologii baz danych jest dziś niepodważalna. Nikt nie kwestionuje podstawowych jej zalet — niezależności programów i danych, wysokiej integralności danych, połączenia z nieproceduralnymi narzędziami dostępu — dzięki którym nastąpiła poprawa jakości usług świadczonych przez ośrodki epd. Wymienione zalety są więc wystarczającym bodźcem do stosowania technologii baz danych (BD), mają jednakże charakter przede wszystkim techniczny. Technologia BD nie spełniła natomiast nadziei organizacji, które widziały w niej narzędzie budowy tzw. wspólnej bazy danych — „zbiornika" kluczowych danych z wielu obszarów dziedzinowych, umożliwiającego budowę zintegrowanego systemu informowania kierownictwa (SIK).

Jak wskazują dostępne źródła [1, 4], prawie wszystkie istniejące bazy danych mają charakter odcinkowy, tj. każda z nich zaspokaja potrzeby nie wykraczające poza jeden obszar dziedzinowy organizacji. Rzadko natomiast można spotkać wspólne bazy danych (WBD), w których zgromadzone są dane z co najmniej dwóch obszarów. Nic więc dziwnego, że systemy zarządzania bazą danych (SZBD) określane są czasem jako „gloryfikowane metody dostępu" lub „kosztowne zabawki dla programistów".

Problemów związanych z wdrażaniem technologii BD jest wiele, natomiast mało miejsca poświęca się im w literaturze — ze zrozumiałych zresztą względów. Pomijając problemy wynikające z niezrozumienia mechanizmów SZBD, mają one charakter przede wszystkim organizacyjny — powodują je najczęściej: złe zarządzanie złożonym przedsięwzięciem, niewłaściwy dobór personelu, zmiany decyzji itp. [5] — są one jednak plagą większości przedsięwzięć informatycznych. Jednocześnie muszą też istnieć jakieś poważne problemy powodujące, że wspólnych baz danych jest tak niewiele. W artykule niniejszym omówione są problemy [1], zrekonstruowane na podstawie dostępnej literatury amerykańskiej, w kontekście integracyjnego oraz ewolucyjnego podejścia do WBD.

PODEJŚCIE INTEGRACYJNE

Niewiele ryzykując można powiedzieć, że w początkowej fazie ekspansji technologii BD podejście integracyjne było jedynym, jakie stosowano. Polega ono na konstruowaniu odcinkowych baz danych (OBD) dla poszczególnych zastosowań, a następnie ich integrowaniu. Jak pisze Palmer [1] „umożliwiało ono szybsze wdrażanie indywidualnych zastosowań ze względu na brak potrzeby [dokładnych] badań powiązań w obrębie bazy danych". Inną przyczyną jego stosowania był zapewne brak znajomości kluczowych elementów podejścia ewolucyjnego. Pomijamy tutaj przypadek, gdy nie planowano integracji OBD ze względu na rzekomy brak istotnych powiązań między danymi z różnych obszarów dziedzinowych organizacji. Jak wykazuje praktyka, założenia te są z reguły nierealne i prędzej czy później występuje potrzeba budowy WBD [5].

Pierwszą OBD budowano zwykle dla relatywnie mniej ważnego zastosowania, bazującego na konwencjonalnej technologii, bez zasadniczego przeprojektowania. Przeważnie powoływano także administratora BD (ABD) o dość skromnych uprawnieniach. Następnie budowano OBD dla kolejnych zastosowań — czasami chaotycznie, żywiołowo i bez pełnej kontroli. Kiedy powstało już wiele OBD, zmaterializowały się znane nam skądinąd problemy. Dane powtarzały się, przez co konieczna była ich wielokrotna aktualizacja. Znajdujące się w OBD dane były co prawda bezpośrednio powiązane, ale brak było powiązań między danymi w różnych bazach danych o odrębnych opisach. W konsekwencji nieproceduralne narzędzia dostępu okazywały się bezradne, gdy zakres danych niezbędnych do wygenerowania odpowiedzi na zapytanie wykraczał poza jedną bazę. Wówczas to przystępowano do planowanej integracji, która jednak okazała się bardzo trudna lub wręcz niemożliwa.

Problemy integracji OBD celem otrzymania WBD można podzielić z grubsza na organizacyjne oraz techniczne. Podstawowym problemem z pierwszej grupy był brak jednolitych organizacyjnych definicji danych. W sytuacji, gdy zastosowania zorientowane były na potrzeby poszczególnych działów organizacji, problem taki w ogóle nie występował. Dopiero przy próbie budowy WBD okazywało się, że użytkownicy z różnych działów stosowali różne interpretacje, nazwy, zasady kodowania i kontroli jakości tych samych w istocie danych, które dotychczas znajdowały się w OBD, a teraz miały być stosowane wspólnie (ewidentny brak słownika danych). Jeżeli więc koncepcja wspólnego stosowania danych na poziomie organizacyjnym miała być realizowana w praktyce, konieczne było ujednolicenie definicji danych. Tymczasem organizacje nie były przygotowane na rozwiązywanie takich problemów. Brakowało mediatora w sytuacjach konfliktowych, tzn. ABD o odpowiedniej pozycji, autorytecie i uprawnieniach [2].

Inny problem stwarzała „własność danych". Niechęć do wprowadzania danych, z których mieliby korzystać także inni użytkownicy, mogła mieć zresztą nie tylko psychologiczne podłoże — być może było to związane z tradycyjnymi zasadami rozliczania kosztów użytkowania systemu komputerowego. Na wspólne stosowanie danych rzutowały także obawy przed uzależnieniem prawidłowego funkcjonowania działu od danych wprowadzanych gdzie indziej — obawy przed błędami i nieterminowością wprowadzania, wynikające z podziału odpowiedzialności. Nie trzeba dodawać, że na rozwiązywanie takich problemów organizacje także nie były przygotowane.

Jeden z głównych problemów technicznych dotyczących integracji OBD był związany z niejednolitością definicji danych na poziomie organizacyjnym. W przypadku budowy odrębnych baz danych przez różne zespoły ludzkie bez dostatecznego porozumiewania się i nadzoru powstawały niekompatybilne programowe definicje danych — wystarczyło, że różniły się nazwy tych samych danych znajdujących się w różnych OBD. Ze względu na zróżnicowanie definicji organizacyjnych, różniły się między sobą także zasady kodowania i kontroli jakości danych, wbudowane w programy. Przy założeniu wspólnego stosowania danych na poziomie programowym, definicje te musiały być jednolite. Ponieważ nie były, pojawiła się konieczność wprowadzania zmian do programów, których zakres zależał od stopnia integracji i

[1] Zrozumienie omawianych problemów wymaga poznania przyczyn i warunków, w jakich powstała koncepcja WBD. Były one przedmiotem kilku artykułów publikowanych w cyklu „Problematyka bazy danych": Mrozik St., Sukiennik J.: Dlaczego wspólna baza danych? INFORMATYKA nr 6/1978, Dubczyński W.: Projektowanie systemu informatycznego w oparciu o wspólną bazę danych. Część 1. INFORMATYKA nr 11/1978; Dubczyński W.: Projektowanie systemu informatycznego w oparciu o wspólną bazę danych. Część 2. INFORMATYKA nr 12/1978, Dubczyński W., Gryc T.: Projektowanie systemu informatycznego w oparciu o wspólną bazę danych. Część 3. INFORMATYKA nr 1/1979

[2] Jak wykazały badania przeprowadzone w 20 organizacjach w Stanach Zjednoczonych, ABD podlegał bezpośrednio kierownikowi działu epd zaledwie w dwóch przypadkach [7]. Niska pozycja zdecydowanej większości ABD nie sprzyjała zabiegom mediacyjnym szczególnie na wyższych szczeblach organizacji

zróżnicowania definicji. Dodatkowym utrudnieniem był niewątpliwie brak zbiorczych informacji o ich lokalizacji w programach, a więc znowu brak słownika danych.

Problemem natury technicznej była także restrukturalizacja wymagająca zmian w programach. Struktury danych w OBD zorientowane były na potrzeby poszczególnych zastosowań. W sytuacji, gdy dane miały być przetwarzane przez programy o różnych wymogach, ich struktury musiały być bardziej uniwersalne. Teoretycznie, przed zmianami programy powinna chronić niezależność programów i danych. Jednakże we współczesnych SZBD jest ona niepełna i w dodatku zróżnicowana. Użytkownicy systemu IMS firmy IBM najwięcej kłopotów mają podobno właśnie z restrukturalizacją, związaną z integrowania OBD, a wynikają one m.in. z ograniczonych możliwości tworzenia struktur sieciowych oraz braku niezależności na poziomie danych elementarnych ([1, 8]).

PODEJŚCIE EWOLUCYJNE

Podejście ewolucyjne do budowy WBD zakłada stopniową rozbudowę jednej bazy tak, aby docelowo zgromadzone były w niej dane z co najmniej dwóch obszarów dziedzinowych organizacji.

Zauważmy, że takie określenie WBD wcale nie implikuje jednej bazy dla wszystkich kluczowych danych organizacji. W istocie bowiem brak zgodności czy budowa takiej WBD jest celowa i/lub możliwa. Wskazuje się np. na problemy integralności i bezpieczeństwa danych związane z daleko posuniętą integracją zastosowań [9]. Z drugiej strony wiadomo, że możliwość podziału bazy danych typu CODASYL na obszary w znacznej mierze problemy te rozwiązuje. Można także spotkać opinię, że w organizacji powinny istnieć dwie WBD — jedna dla zastosowań operatywnych i druga (wyspecjalizowana) dla SIK [10]. Problem zakresu WBD należy więc zostawić otwarty.

Podejście ewolucyjne zaczęto stosować stosunkowo niedawno i to w nielicznych przypadkach. Ma ono pewne charakterystyczne elementy, których realizacja jest w owym podejściu konieczna. Powstały one w znacznej mierze na bazie doświadczeń podejścia integracyjnego, aczkolwiek niektóre z nich od pewnego czasu były sygnalizowane przez teoretyków. Podstawowe elementy tego podejścia to planowanie ewolucji bazy danych, traktowanie danych jako zasobu należącego do całej organizacji, administrowanie danych (pojęcie szersze niż administrowanie BD) oraz ich modelowanie. Planowanie ewolucji uważa się za antidotum na żywiołowe powstawanie OBD, nie zawsze w logicznej kolejności i nie zawsze tam gdzie jest to celowe.

Planowanie ewolucji należy więc rozumieć jako określenie ogólnych ram rozwoju bazy danych — określenie etapów rozwoju i zakresu WBD. Uważa się to za konieczne ze względu na długotrwałość przedsięwzięcia (5—10 lat) oraz dużą liczbę zastosowań, w których wykorzystuje się konwencjonalną technologię epd. Traktowanie danych jako zasobu należącego do całej organizacji i przez nią wykorzystywanego ma być punktem wyjścia do rozwiązania problemów wspólnego stosowania danych. Mamy więc do czynienia z założeniem wspólnego stosowania a priori, z którego wynika ujednolicenie organizacyjnych definicji danych.

Kolejnym elementem podejścia ewolucyjnego jest administrowanie (zarządzanie) zasobem danych zarówno na poziomie organizacyjnym, jak i programowym, pod kątem stopniowego umieszczania ich w WBD zgodnie z zaplanowaną ewolucją. W tym celu niezbędne jest utworzenie grupy (lub grup) ludzi o obszernym zakresie funkcji (obejmującym także klasyczne funkcje ABD), mających odpowiednią pozycję i uprawnienia. Ze względu na dużą liczbę typów danych za narzędzie niezbędne dla administrowania danych uważa się komputerowy słownik danych. Wreszcie, dane należy modelować w oderwaniu od zastosowań — tak, aby ich struktury były „naturalne" i odzwierciedlały bardziej rzeczywistość niż potrzeby zastosowań.

Praktyczna realizacja omówionych pokrótce elementów podejścia okazała się jednak trudna ze względu na problemy organizacyjne i metodologiczne. Sukces przedsięwzięcia zależy przede wszystkim od poparcia kierownictwa dla wspólnego stosowania danych. Poparcie to uzależnione jest z kolei od zrozumienia nie tylko zalet WBD, ale także jej wymogów, z czym nie zawsze jest najlepiej.

Organizacje nadal nie traktują danych jako zasobu, być może dlatego, że zakres danych zgromadzonych w pamięciach zewnętrznych komputerów nie obejmuje wszystkich danych organizacji, a jedynie ich część (20—25%) znajduje się w bazach danych. W konsekwencji koncepcja wspólnego stosowania danych nie znajduje posłuchu — nie ma zrozumienia i chęci do przezwyciężenia problemów oraz postaw opisanych poprzednio. Brak poparcia dla wspólnego stosowania danych powoduje, że powstają OBD [4, 11].

Szczególnie niebezpieczne jest niedocenianie lub negacja potrzeby ujednolicenia organizacyjnych definicji danych, ze względu na brak rozeznania co do ilości typów danych, stopnia zróżnicowania definicji i czasu potrzebnego na ich ujednolicenie. Tymczasem liczba typów danych jest bardzo duża, a standaryzacja definicji wymaga sporo czasu. Canning w [12, 13] podaje przykłady organizacji, z których jedna zidentyfikowała aż 700 (!) typów danych charakteryzujących pracownika, natomiast wysiłek, jaki włożono w identyfikację oraz ujednolicenie wszystkich organizacyjnych definicji danych w drugiej organizacji, stanowił równowartość około 30 osobolat.

Brak spojrzenia na dane organizacji jako na zasób wymagający zarządzania powoduje także brak zrozumienia istoty i celu administrowania danymi. Wyraża się on w niechęci do tworzenia nowych stanowisk, przesunięć uprawnień i funkcji, w lokalizacji administratora na niższych szczeblach działu epd, a także w traktowaniu komputerowego słownika danych jako zbędnego dodatku do SZBD. Dodatkowym utrudnieniem jest brak zgodności wśród teoretyków i praktyków co do podziału funkcji związanych z zarządzaniem zasobu danych. Dlatego też istnieje mnóstwo koncepcji dotyczących podziału funkcji planistycznych, projektowych i operatywnych między administratorów na różnych szczeblach organizacji.

Oprócz organizacyjnych problemów budowy WBD w oparciu o podejście ewolucyjne istnieją jeszcze problemy metodologiczne. Nie ma bowiem — jak dotychczas — pełnej, sformalizowanej metodologii ewolucyjnej budowy WBD, określającej dokładnie co, kiedy, kto i jak. Co gorsza, teoria nie wypracowała jeszcze żadnej metodologii budowy bazy danych, ponieważ nie ma formalnych narzędzi umożliwiających płynne przejście od rezultatów analizy i założeń zastosowania poprzez modelowanie rzeczywistości i danych do projektowania struktur BD. Istnieje wiele (czasami aż nadto) rozwiązań wycinkowych (np. w dziedzinie modelowania danych), nie wbudowanych jednak w strukturę całego przedsięwzięcia i bez powiązań z innymi etapami.

Problemom projektowania baz danych poświęca się także podejrzanie mało miejsca w podręcznikach technologii baz danych lub — co paradoksalne — w podręcznikach projektowania BD. Tak więc teoria budowy baz danych nie spotkała się jeszcze z praktyką. Dlatego też praktycy zdani są przede wszystkim na własny zdrowy rozsądek, intuicję i doświadczenie. I nieźle sobie radzą, korzystając z tych — jakże nieuchwytnych — narzędzi...

Palmer wspomina [14] o metodologii budowy baz danych — opartej na bogatych wieloletnich doświadczeniach — stosowanej z powodzeniem przez firmę konsultacyjną CACI. Należy podejrzewać, że metodologii takich jest więcej.

* * *

Problemy związane z budową WBD są wynikiem braku zrozumienia wymogów i potencjalnego wpływu wspólnego stosowania danych na organizacje, a nie z samej technologii BD. Problemy organizacyjne, oscylujące wokół wspólnego stosowania danych na poziomie organizacyjnym i programowym, są wspólnym mianownikiem dwóch omówionych podejść. W integracyjnym — utrudniały lub uniemożliwiały integrację OBD, w ewolucyjnym — utrudniały lub uniemożliwiały rozpoczęcie prac nad WBD. Problemy te nie występowały oczywiście wszędzie i z jednakowym natężeniem (zależne są bowiem od czynników zróżnicowanych w każdej organizacji: od profilu, wielkości, terytorialnego rozproszenia, stopnia decentralizacji zarządzania itp.). Omówienie ich w kontekście dwóch podejść nie oznacza, że są one jedynymi drogami do WBD. W literaturze można nawet zauważyć propagowanie podejścia mieszanego — integracyjnego ze stosowaniem komputerowego słownika danych. Postuluje się mianowicie instalowanie słownika p r z e d

zainstalowaniem SZBD, celem uporządkowania najpierw organizacyjnych i programowych definicji danych w zastosowaniach konwencjonalnych, a następnie standaryzacji definicji danych w OBD.

W związku z oczekiwaną stopniową ekspansją technologii BD w kraju warto zadać pytanie czy problemy omówione w artykule będą identyczne. Zdaniem autora — tak, o ile oczywiście WBD będzie celem zastosowania technologii BD. Nie powinny wystąpić w takim natężeniu jedynie problemy restrukturalizacji, gdyż krajowy system RODAN zapewnia wyższy poziom niezależności programów i danych niż np. IMS. Pozostałych problemów nie można jednak bagatelizować. W organizacjach brakuje bowiem w ogóle definicji danych, nie mówiąc już o jednolitych definicjach danych stosowanych w różnych obszarach dziedzinowych. Jakże często poza tym organizacje nie mają pojęcia o zawartości dziedzinowych kartotek manualnych, o rozmiarach i zróżnicowaniu zasobu danych! Niewiedza taka utrudni — lub uniemożliwi — rozpoczęcie i kontynuowanie wszelkich działań dotyczących WBD.

O realności problemów wspólnego stosowania danych na poziomie organizacyjnym i programowym autor przekonał się osobiście, biorąc udział w przedsięwzięciu zakładającym integrację na poziomie zbioru dwóch zastosowań z różnych obszarów dziedzinowych organizacji. Wystąpiły wówczas — oczywiście w mniejszej skali — prawie wszystkie problemy wspólnego stosowania danych omówione w artykule, mimo że nie korzystano z technologii BD.

Konieczne wydaje się nie tylko propagowanie zalet i możliwości technologii BD, ale także problemów związanych z jej wdrażaniem oraz jej wymogów, zwłaszcza wspólnego stosowania danych. Tą drogą uniknie się z pewnością wielu rozczarowań i niepowodzeń. Pożądane jest także rozpoczęcie prac nad komputerowym słownikiem danych — zanim nie stanie się on potrzebą aż nadto palącą.

LITERATURA:

[]1 Palmer I.: Database Systems. A Practical Reference. CACI International 1975

[2] Boot R.: Doświadczenia użytkowników stosujących systemy gospodarowania bazami danych — w: „Baza danych, Europejski Program Badawczy Diebolda", Zeszyt 72. OBRI 1975, s. 104—130

[3] Davis B.: User Experience with DBMS in the United Kingdom. DATA PROCESSING DIGEST 1978/4, s. 18—19

[4] Wiorkowski G., Wiorkowski J.: Does a Database Management System pay off? DATAMATION 1978/4, s. 109—114

[5] Sibley E. H.: The Impact of Database Technology on Business Systems — w: „Information Processing 77". North-Holland Publishing Co., 1977, s. 589—596

[6] Fry J. P., Sibley E. H.: Evolution of Database Management Systems. ACM COMPUTING SURVEYS 1976/1, s. 7—42

[7] DeBlasis J., Johnson T. H.: Data base administration. Classical pattern, some experiences and trends — w: „Proceedings of the 1978 National Computer Conference". AFIPS Press 1978, s. 1—7

[8] Curtice R. M.: Data Independence in Database Systems. DATAMATION 1975/4, s. 65—71

[9] Sibley E. H.: The Development of Database Technology. ACM COMPUTING SURVEYS 1976/1, s. 1—5

[10] Martin J.: Computer Data Base Organization. Prentice-Hall, Inc. 1974

[11] Slesser R. M.: One County's Experience with the Data Base. DATAMATION 1979/2, s. 183—186

[12] Canning R.: The Data Administrator Function. ADP ANALYZER 1972/11

[13] Canning R.: The Cautious Path to a Data Base. EDP ANALYZER 1973/6

[14] Palmer I.: Comments on Dr. Sundgren's Paper: Database Design in Theory and Practice — Towards an Integrated Methodology — w: „Issues in Data Base Mangement". North-Holland Publishing Co. 1979, s. 29—30

LESZEK A. MACIASZEK

Akademia Ekonomiczna
Wrocław

Reorganizacja baz danych

Szybki rozwój badań i zastosowań technologii baz danych — zapoczątkowany raportami Komitetu CODASYL z pierwszej połowy lat sześćdziesiątych — rodzi określone konsekwencje w konstrukcji, technologii i oprogramowaniu systemów informatycznych. W Polsce pierwszym wyrazem tego nowego spojrzenia na rozwój metod informatycznych było opracowanie Uniwersalnego Systemu Zarządzania Bazą Danych RODAN na komputery JS EMC. System RODAN dostarcza mechanizmów programowych pozwalających na założenie i eksploatację systemów informatycznych opartych na bazie danych. W ostatnich latach odpowiednie prace projektowo-programowe zostały podjęte przez wiele ośrodków w kraju, między innymi w Akademii Ekonomicznej we Wrocławiu, gdzie prowadzi się prace na rzecz KM Huta Katowice (ponadto system RODAN jest wykorzystywany w badaniach naukowych i w procesie dydaktycznym).

W ślad za pierwszymi zastosowaniami technologii bazy danych okazało się, że użytkownik wymaga od systemu zarządzania bazą danych nie tylko mechanizmów pozwalających na założenie i eksploatację bazy danych, ale także mechanizmów umożliwiających jej rozwój (a więc dokonywanie zmian, usuwanie i wprowadzanie nowych elementów i relacji, usuwanie redundancji, zmianę charakterystyk danych itp.). Systemy bazy danych [1] mają bowiem sens tylko wówczas, gdy rozumiane są jako obiekty dynamiczne, łatwo modyfikalne zarówno na poziomie struktury logicznej, jak i fizycznej. Stąd też twórcy systemów zarządzania bazą danych, w tym autorzy systemu RODAN [3, 4], podjęli prace mające na celu dostarczenie odpowiedniego oprogramowania. W nomenklaturze baz danych nosi ono nazwę procesora reorganizacji [2].

Termin **reorganizacja bazy danych obejmuje** — przynajmniej w rozumieniu tego artykułu — zarówno zmiany w zakresie struktury logicznej bazy danych (ang. *restructuring*), jak i modyfikacje w strukturze fizycznej (ang. *reorganization*). Te dwa rodzaje reorganizacji będziemy nazywać odpowiednio reorganizacją logiczną [3] i reorganizacją fizyczną.

REORGANIZACJA LOGICZNA BAZY DANYCH

Reorganizacja logiczna dotyczy związków logicznych między danymi i rekordami w bazie danych. Innymi słowy daje ona możliwość uzupełnienia lub poprawienia wszystkich pierwotnych wyrażeń Języka Opisu Danych, składających się na początkowy schemat bazy danych. W szczególności zmiany te mogą dotyczyć [1, 2, 3, 4]:

[1] Przez „system bazy danych" rozumie się system informatyczny projektowany i eksploatowany w technologii bazy danych

[2] W przypadku systemu RODAN oprogramowanie to znane jest jako procesor R/R

[3] W literaturze polskiej można spotkać nazwy równoważne, mianowicie: restrukturyzacja, restrukturacja i rewizja struktury

— usunięcia typu danej, typu rekordu, typu zbioru lub typu obszaru
— dodania nowego typu danej do istniejącego typu rekordu
— dodania nowego typu rekordu do istniejącego lub nowego typu obszaru
— zmiany struktury wewnątrzrekordowej (np. poprzez ustanowienie z określonej grupy danych nowego typu rekordu lub poprzez przeniesienie danej z jednego typu rekordu do innego)
— zmiany charakterystyki typu danej (np. w zakresie podstawy, sposobu przedstawienia lub precyzji)
— zmiany opisu typu zbioru (np. włączenie nowego typu rekordu składowego do istniejącego typu zbioru, wprowadzenie nowego typu zbioru łączącego nowe lub istniejące typy rekordów, zmiana uporządkowania rekordów składowych)
— zmiana nazwy typu danej, typu rekordu, typu zbioru lub typu obszaru.

Decyzja reorganizacji logicznej bazy danych stanowi konsekwencję żądanych zmian w zawartości informacyjnej w bazie danych. Zakres tych zmian należy uprzednio poddać wnikliwej analizie, ażeby uchronić programy użytkowe przed koniecznością zbyt dużych modyfikacji.

Reorganizacja logiczna zawsze wywołuje określone zmiany w strukturze fizycznej bazy danych. Natomiast modyfikacje w strukturze fizycznej (reorganizacja fizyczna) nie muszą powodować zmian w strukturze logicznej. Oznacza to, że organizacja fizyczna może mieć charakter czysty (bez zmian schematu) lub też może być pochodną zmian w schemacie (przede wszystkim w deklaracjach Języka Opisu Pamięci [4]). Niekiedy reorganizacja o charakterze czystym nazywana jest reorganizacją rozmieszczenia fizycznego (ang. *physical placement reorganization*), natomiast reorganizacja wynikająca ze zmian w opisie schematu — reorganizacją strategiczną (deklaratywną) (agn. *strategy reorganization*) [1].

REORGANIZACJA FIZYCZNA BAZY DANYCH

W procesie reorganizacji fizycznej następuje przemieszczenie wystąpień rekordów w bazie danych mające na celu optymalizację dostępu do danych. Administrator bazy danych ma możliwość uzupełnienia lub poprawienia zdań Języka Opisu Pamięci, składających się na pierwotny opis bazy danych. Reorganizacja strategiczna może dotyczyć [1, 2, 3, 4]:

- typu obszaru
  — przypisanie typu obszaru do innego zbioru fizycznego, tj. do zbioru o innej nazwie *dd (data definition)* Języka Opisu Zadań — może to pociągnąć za sobą zmianę typu woluminu pamięci zewnętrznej, jak również zmianę charakterystyk zbioru
  — zmiany rozmiaru obszaru
  — zmiany wielkości stron danego obszaru
  — zmiany współczynnika wypełnienia stron podczas pierwotnego ładowania bazy danych
  — zmiany ilości łańcuchów synonimów (tj. łańcuchów CALC) dla strony

- typu rekordu
  — zmiany rozmieszczenia rekordów w obszarze (np. w pobliżu rekordu macierzystego lub z określonym interwałem wyrażonym w stronach)
  — zmiany w zakresie opcji kompresji danych na stronie
  — zmiany w zakresie ilości przeszukiwanych stron w procesie szukania wolnego miejsca dla zapisu nowych rekordów w bazie

- typu zbioru
  — zmiany w zakresie sposobu fizycznej realizacji zbioru (łańcuchowy lub tablicowy)
  — zmiany sposobu przechowywania indeksów (np. w pobliżu lub w oddaleniu od rekordu macierzystego)
  — zmiany charakterystyk segmentów stron zawierających indeksy
  — zmiany rodzajów realizowanych powiązań między rekordami składowymi oraz między rekordami składowymi a rekordem macierzystym (dotyczy dodatkowych możliwości powiązań do przodu oraz powiązań z rekordem macierzystym).

Procesor reorganizacji powinien również stwarzać możliwość ingerencji administratora bazy danych w strukturę fizyczną bez zmian w schemacie (reorganizacja rozmieszczenia fizycznego). Tego typu ingerencje dotyczą zazwyczaj wystąpień rekordów i mogą objąć czynności:
— przeniesienia wystąpień rekordu do innego obszaru (jednakże w ramach obszarów wyspecyfikowanych w klauzuli WITHIN)
— optymalizacji indeksów, tablic i łańcuchów synonimów
— zmiany rozmieszczenia rekordów w zbiorze (zwykle chodzi o zagęszczenie)
— zmiany uporządkowania rekordów w zbiorze (w celu osiągnięcia kolejności fizycznej rekordów zgodnej z ich kolejnością logiczną) oraz w obszarze (zgodnie z deklarowanym porządkiem).

PROCES REORGANIZACJI

Efektem procesu reorganizacji jest przeniesienie wartości danych ze starej bazy do nowej. Nie jest to zatem nic innego jak zakładanie (ładowanie) nowej bazy danych. Oczywiście czynność ta ma swoją specyfikę, określoną wszystkimi poprzednimi etapami procesu reorganizacji. W szczególności podczas ładowania nowej bazy danych duże znaczenie ma sposób przygotowania do tej czynności bazy starej. Stąd też w procesie reorganizacji można wyodrębnić dwie fazy:
— fazę przygotowawczą oraz
— fazę właściwą.

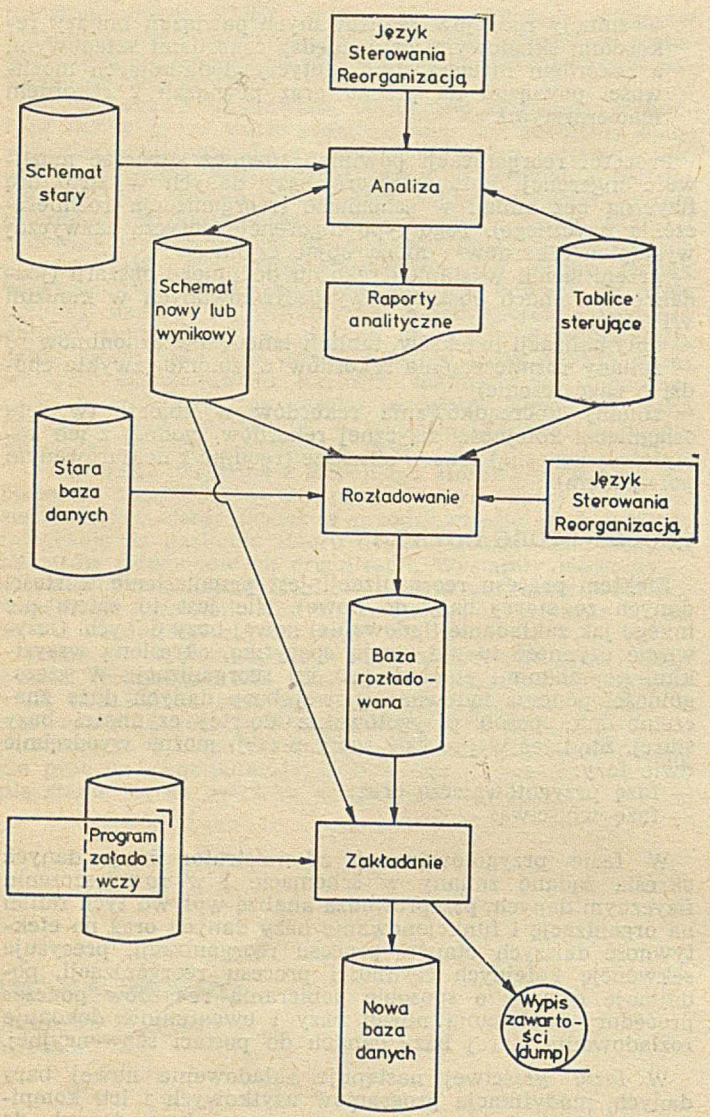
W fazie przygotowawczej administrator bazy danych określa żądane zmiany w schemacie i w rozmieszczeniu fizycznym danych, przeprowadza analizę wpływu tych zmian na organizację i funkcjonowanie bazy danych oraz na efektywność dalszych etapów procesu reorganizacji, precyzuje sekwencję kolejnych czynności procesu reorganizacji, podejmuje decyzję o sposobie pobierania rekordów podczas procedury zakładania nowej bazy i ewentualnie dokonuje rozładowania starej bazy danych do postaci sekwencyjnej.

W fazie właściwej następuje załadowanie nowej bazy danych, modyfikacja programów użytkowych i ich kompilacja, testowanie i udostępnienie nowej bazy danych do eksploatacji. W fazie tej istnieje możliwość wykorzystania procesora reorganizacji w sposób dający dodatkowe korzyści. Przykładowo można tak sterować przebiegami oraz starym i nowym schematem, aby łączyć bazy niezależne lub rozdzielać (łączyć) fragmenty baz danych [3].
W zależności od konkretnej implementacji i od zakresu reorganizacji istnieje wiele sposobów osiągnięcia żądanych zmian. W szczególności procesor reorganizacji może mieć postać mniej lub bardziej rozbudowaną. W wersji złożonej może on spełniać nawet takie funkcje, jak [1, 3]:
— generowanie raportów wskazujących na kontaminację proponowanych zmian
— szacowanie czasu i zasobów koniecznych do reorganizacji
— generowanie listy możliwych niekonsekwencji i błędów w fazie właściwej
— tworzenie nowej wersji schematu
— półautomatyczna modyfikacja programów użytkowych.

Na procesor reorganizacji składają się zazwyczaj translator Języka Sterowania Reorganizacją (ang. *Reorganization Control Language*) [5], zdania sterujące przebiegiem procesora oraz procedury standardowe realizujące czynności, które dają się zautomatyzować. W fazie przygotowawczej jest wykorzystywany przede wszystkim Język Sterowania Reorganizacją, natomiast w fazie właściwej — procedury standardowe. Działania składające się na proces reorganizacji zostały pokazane w formie graficznej na rysunku poniżej. W zależności od implementacji i zakresu dokonywanych zmian możliwe są inne przebiegi procesora reorganizacji.

[4]) Znaczna część systemów zarządzania bazą danych opartych na koncepcji Komitetu CODASYL (w tym RODAN i DMS-1100) posiada wspólny kompilator dla Języka Opisu Danych Schematu i dla Języka Opisu Pamięci — stąd też pojęcie schematu dotyczy zarówno logicznego, jak i fizycznego opisu bazy danych

[5]) Istnieje kilka nazw alternatywnych, np. Data Strategy Change Language [1], Data Structure Translation Language [3, 4]



Proces reorganizacji baz danych

STRATEGIA REORGANIZACJI

Administrator bazy danych ma możliwość różnorodnego sterowania przebiegiem reorganizacji. Między innymi może on podzielić przebieg procesora na autonomiczne etapy za pomocą punktów kontrolnych. Dla każdego etapu administrator powinien określić zakres reorganizacji oraz wymagane procedury odtwarzania stanów bazy danych i zachowania integralności bazy. Przerwany przebieg procesora reorganizacji wznawia się za pomocą odpowiednich zdań sterujących w czasie określonym przez administratora.

Strategia reorganizacji jest uzależniona także od wielu innych czynników, z których najważniejsze to [1, 2, 5]:
— zasięg procesu reorganizacji (szczególnie ważne jest określenie granic i warunków reorganizacji, gdy nie jest wymagana reorganizacja całej bazy danych)
— konieczność zachowania równoległego dostępu do części bazy danych wyłączonych spod reorganizacji
— czas przeznaczony na reorganizację
— klasa i tryb eksploatacji systemu bazy danych podlegającego reorganizacji.

Z punktu widzenia ostatniego z wymienionnych czynników istnieje możliwość przeprowadzenia procesu reorganizacji [1]:

— równolegle z normalną eksploatacją bazy danych, przez ustanowienie do czasu zakończenia reorganizacji wyłącznego sterowania obszarami, rekordami i urządzeniami pamięci, których reorganizacja dotyczy (przykładem użycia tej metody może być procedura, podczas której należy skasować typ rekordu i w ślad za tym usunąć odpowiednie typy zbiorów oraz zmienić charakterystykę właściwego obszaru)

— dynamicznie podczas normalnej eksploatacji bazy danych (przykładowo, jeśli ulegają zmianie atrybuty podstawy danych — np. z dziesiętnego na binarny — to istnieje możliwość zmodyfikowania odpowiednich wystąpień rekordów podczas normalnej eksploatacji bazy danych)

— w czasie, gdy system bazy danych nie jest eksploatowany.

Na zakończenie tych rozważań warto zwrócić uwagę na fakt, że wskazane na rysunku etapy rozładowania bazy danych do postaci sekwencyjnej i ponownego jej załadowania nie zawsze są konieczne. Niekiedy pożądane zmiany w fizycznej bazie danych można osiągnąć przez zastosowanie innych metod — znanych z procesów aktualizacji danych w zbiorach dyskowych (niekoniecznie organizowanych w technologii bazy danych). Chodzi tu o metodę nakładania (np. przy reorganizacji wybranych wystąpień rekordów lub zbiorów) i kopiowania (np. reorganizacja rozmieszczenia wystąpień rekordów możliwa jest przy przenoszeniu części lub całości bazy do innej bazy danych).

* * *

Wraz z rozwojem teorii i zastosowań technologii bazy danych szczególnego znaczenia nabierają procedury reorganizujące struktury logiczne i fizyczne danych w bazach istniejących. Stąd też użytkownicy systemów zarządzania bazami danych domagają się od ich producentów procesorów reorganizacji, pozwalających na przeprowadzenie odpowiednich modyfikacji z zachowaniem integralności i logiczności baz. W praktyce procesory reorganizacji mogą spełniać wiele funkcji, niekiedy znacznie odbiegających od właściwego rozumienia terminu „reorganizacja". W efekcie procesory te przyjmują charakter pakietów standardowych, bardzo przydatnych administratorowi bazy danych w pracach mających na celu optymalizację, konserwację, zachowanie aktualności i integralności bazy danych. Dlatego dobrze się stało, że przed paru laty twórcy systemu RODAN podjęli prace nad procesorem reorganizacji. Dzięki temu system RODAN powinien w najbliższym czasie zaspokoić wymagania użytkowników także w kwestiach, które są przedmiotem niniejszego artykułu.

LITERATURA:

[1] B.C.S./CODASYL DDLC. Data Base Administration Working Group Report. British Computer Society 1975
[2] Data Reorganization Utility — w: „Sperry Univac 1100 Series. DMS 1100 Level 7R1 System Support Functions". Sperry Rand Corporation 1978
[3] Niewolski S.: Reorganizacja i rewizja struktury danych w SZBD RODAN — w: „Systemy zarządzania bazami danych i przegląd zastosowań oprogramowania komputerów Jednolitego Systemu RIAD". Kursokonferencja naukowa. Kalisz TNOiK 1979
[4] Niewolski S.: Restructure and reorganization of data base in Universal Data Base Management System — RODAN (maszynopis powielany)
[5] Shneiderman B.: Optimum data base reorganization points. Communications of the ACM 1973, nr 6

Łamy INFORMATYKI otwarte dla wszystkich!

Zanim jednak nasi Autorzy sięgną po pióro, prosimy, by zechcieli zapoznać się z poniższymi informacjami.

Nadsyłane artykuły nie mogą być publikowane lub przeznaczone do opublikowania w innych czasopismach.

W artykułach można omawiać, prezentować lub proponować wszystko, co dotyczy współczesnej informatyki, oraz wszystko, co wiąże się z jej kierunkami rozwoju — zarówno z pozycji informatyka, jak i użytkownika informatyki.

Materiał, oprócz tekstu zasadniczego, powinien zawierać — na oddzielnych stronach — kartę tytułową (strona 1), krótki życiorys zawodowy autora (strona 2) i jego zdjęcie, wykaz literatury, tabele, rysunki, podpisy pod rysunki, zdjęcia.

Na stronie 1 należy podać tytuł naukowy, imię i nazwisko, nazwę zakładu pracy, adres prywatny i telefon, tytuł artykułu oraz informację, jaką drogą przesłać honorarium po opublikowaniu artykułu: kasa Wydawnictwa, poczta, bank (w takim przypadku prosimy podać numer konta PKO).

Konstrukcja artykułu powinna być zwarta i przejrzysta; wstęp musi wprowadzić czytelnika w zagadnienie, w podsumowaniu należy sformułować wnioski; podział na rozdziały, podrozdziały i akapity powinien być logiczny i konsekwentny. Należy zwrócić szczególną uwagę na poprawność stylistyczną i terminologiczną, unikać skrótów, rzadko stosowanych wyrażeń obcych i żargonu fachowego; starannie definiować nowe terminy. Należy również wystrzegać się nieczytelnych i zbyt rozbudowanych wzorów.

Tekst powinien być napisany na maszynie, jednostronnie, na papierze nieprzebitkowym formatu A-4, z marginesem 5 cm (30 wierszy na 1 stronie, 60 znaków w 1 wierszu).

Wykaz literatury powinien zawierać: kolejny numer pozycji (w nawiasie kwadratowym), nazwisko i imię autora, tytuł publikacji (książki lub artykułu), ewentualnie tytuł i numer czasopisma (w przypadku artykułu), miejsce i rok wydania.

Tabele — każda na oddzielnej stronie — powinny być numerowane i opatrzone tytułem oraz ściśle związane z tekstem (odniesienie na marginesie).

Rysunki — każdy oddzielnie (uwaga: nie wklejać rysunków w tekst!) — powinny być czytelne i również ściśle związane z tekstem (odniesienie na marginesie). Format rysunku nie może być mniejszy niż 10 × 10 cm.

Podpisy pod rysunkami, napisane również na oddzielnej stronie, oprócz kolejnego numeru powinny zawierać tytuł rysunku i ewentualnie legendę dotyczącą poszczególnych elementów.

Łączna objętość materiału nie powinna przekraczać w przypadku

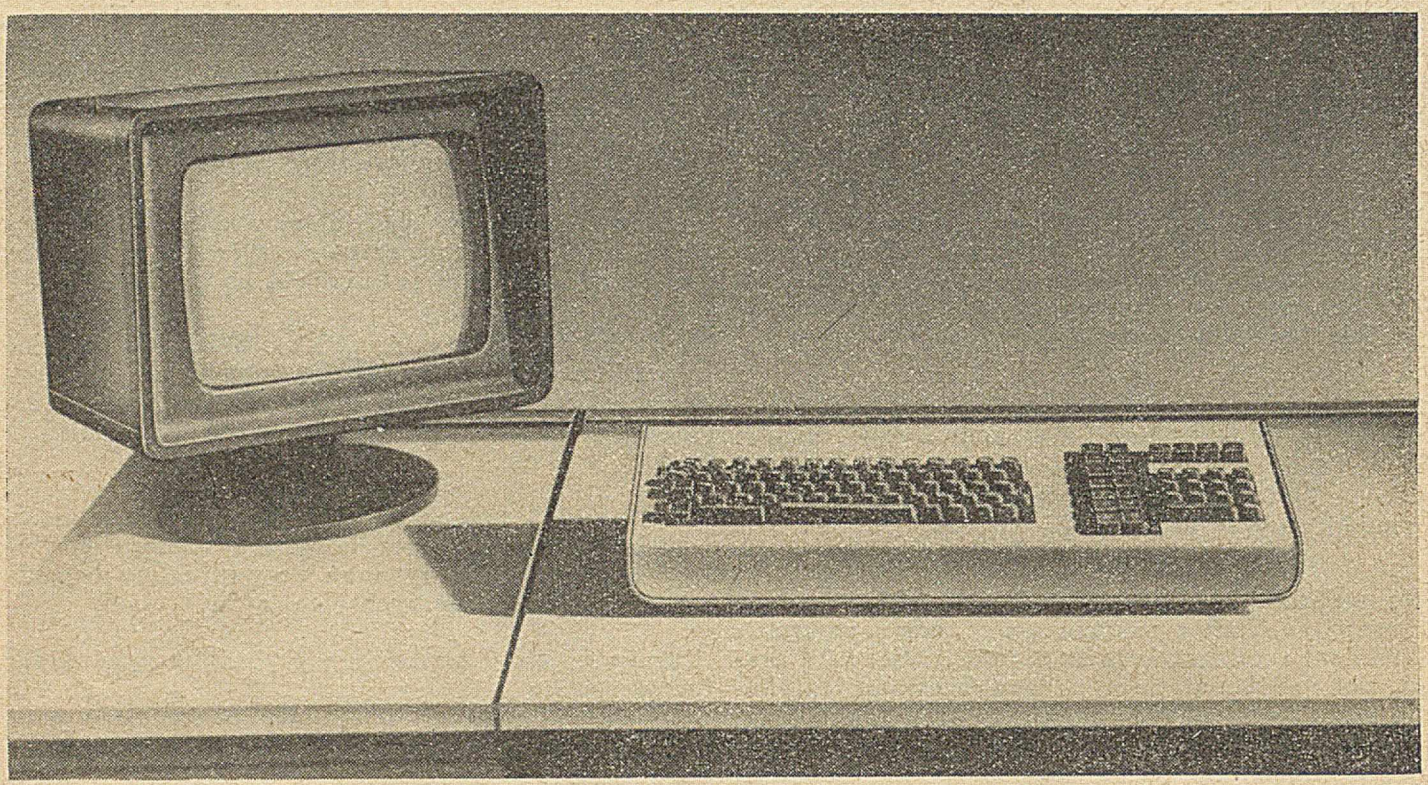
- artykułu problemowego — 12 stron
- reportażu — 8 stron
- recenzji, relacji z imprezy — 6 stron
- informacji — 4 stron maszynopisu

Tak przygotowany materiał prosimy dostarczyć w dwóch egzemplarzach pod adresem: redakcja INFORMATYKI, ul. Jasna 14/16, 00-041 Warszawa. Wszelkich dodatkowych informacji udzielamy pod telefonem 27-71-40.

Autor opublikowanego w INFORMATYCE artykułu otrzymuje bezpłatnie egzemplarz okazowy.

Materiałów nie zakwalifikowanych do druku redakcja nie zwraca.

robotron 4230



Za pomocą tego systemu gromadzenia danych można szybko i efektywnie rejestrować dane

System gromadzenia danych robotron 4230 można stosować zarówno w sposób scentralizowany, jak i rozproszony, w którym poszczególne stacje rejestracji danych zainstalowane są w różnych miejscach. Tam też odbywa się rejestracja wszystkich danych niezbędnych do szybkiego opracowania zleceń i ich terminowej realizacji. Rozwiązanie to jest szczególnie przydatne w handlu i instytucjach usługowych. Wszędzie tam, gdzie konieczna jest rejestracja znacznej liczby danych. System gromadzenia danych robotron 4230 oznacza centralne gromadzenie danych wprowadzonych przez pojedyncze stacje rejestracji danych do pamięci pośredniej, oznacza sterowane komputerem przygotowanie i wstępne przetwarzanie danych, oznacza wprowadzanie danych na kompatybilną komputerową taśmę magnetyczną z przekazywaniem tych danych do komputera w trybie *on-line*.

Podczas rejestracji danych wykonywane są liczne, sterowane komputerem operacje kontrolne. Dzięki nim osiąga się wysoki stopień dokładności, a jednocześnie odciąża komputer od wielu zadań kontrolnych. Udzielamy szczegółowych informacji na ten temat.

System gromadzenia danych robotron 4230 to racjonalne i bezbłędne rejestrowanie danych!

Ambasada NRD
Biuro Radcy Handlowego
Wydział Polityki Handlowej
Dział Informatyki i Maszyn Biurowych
Aleja I Armii Wojska Polskiego 2/4
01-524 Warszawa

robotron

bierze udział w Międzynarodowych Targach Lipskich od 9 do 15 marca br. (wystawa: hala 15, Biuro Eksportu i Importu: hala 12.12)
Biuro Eksportu i Importu: hala 12.12

Robotron Export-Import
Przedsiębiorstwo Handlu Zagranicznego
Niemieckiej Republiki Demokratycznej
DDR 108 Berlin, Friedrichstrasse 61

EO/1571/K/79